令和 2 年
卒業論文

SNS における誹謗中傷文を判別するシステムの提案

指導教員
岡村　真吾

提出日:2021 年 2 月 24 日

奈良工業高等専門学校　情報工学科
5年　1番
池松　新

誹謗中傷文を判別するシステムを有するSNSの提案

岡村研究室　池松　新

SNS上における誹謗中傷によって追い込まれ死を選ぶ人や，心に傷を負う人の割合が近年急増している．本研究ではそのような背景を踏まえ，SNSシステムにおける誹謗中傷文をより正確に判別するシステムについて新規手法を提案した．既存手法の新語・造語に対応できない点やSNSのような短文の個人ブログ特有の言い回しなどに対応できない現状を解決するために，まずデータセットを巨大掲示板からTwitterに変更し，既存研究では形態素解析を用いる際にMeCabとその付属の辞書をそのまま使用していたところを，MeCab Ipadic Neologdに変更した．そしてSO-PMIを用いて単語ごとに悪口かどうかを判別していたが，本研究においては文全体をニューラルネットワークを用いて判断することで新語，造語自体が悪性かどうかわからない場合でも，関連語や，文一文を学習させることで悪性かどうかを判断できると考えた．その考えをもとに，誹謗中傷文を判別するシステムの実験を行い，その実験方法としてGoogle Colaboratory上にプログラムを構築し，実装を行った．そして実装後，実験結果としてクラスの動作確認やMeCab-Ipadic-Neologdを用いていることで新しい言葉などにも対応できているかどうかの動作確認を行った後，テストデータをもとに予測したラベルデータの精度を求めた．ラベルデータの精度としては91.66%を出したものの，悪性のデータセット数が少ないことやデータに偏りがあることなどから，実際の悪性の文章の判別率は50%程度となった．

また本研究において今後の課題として本人確認を行う新たなSNSとして提案しており，匿名性を維持しつつ，疑わしい投稿にはSNSの表現の自由の維持しながら個人情報を記録しておく方法の提案をし，このような現状のSNS課題点として挙げられていた点を5.2節で述べたポイントを踏まえて対策案を提案することができた．

目次

第 1 章　まえがき

近年様々な著名な方々が自ら死を選ぶ事態が相次いでいる. 2020 年コロナ禍において世情が大きく変化し精神的に不安定になったことも一つの理由であると考えられるが, その他の大きな原因として SNS 上において自由な発信ができることが出来るが故に起きる個人, 団体に対しての誹謗中傷が絶えない点である. そしてたくさんの人が心無い言葉によって傷ついている現状がある. そのようなSNSが普及した現代社会の社会問題を解決するべきだと考えた.

SNS(ソーシャルネットワークサービス)とは, インターネットを介して人間関係を構築できるスマホ・パソコン用の Web サービスの総称であり, 電子掲示板などとは違い, SNS では特に「情報の発信・共有・拡散」といった機能に重きを置いているのが特徴である. そしてその SNS 上における誹謗中傷によって追い込まれ死を選ぶ人が存在する現実がある. また, 死まで追いやられない場合においても, 心に傷を負う人の割合が近年徐々にではあるが増加している現実(図 1.1)がある. また, SNS を代表する Twitter 社もスタッフを動員して誹謗中傷対策(通報された内容を主に対応)を行っているが, 追い付いていない技術的に困難な現状がある.

図 1.1 誹謗中傷による警察相談件数[1]

本研究では, 研究背景においてあげたような現状を問題視しており誹謗中傷文がインターネット上に流れることを防止, また万が一そのような被害が生まれた時にいち早く誹謗中傷を行った人物を特定するために「誹謗中傷文を判別し本人確認を行う新たなSNS」を本研究において提案する.

そして本研究では, SNS の投稿文判別システムにおいて誹謗中傷文をニューラルネットワークを用いて判別するシステムについて研究する. 本論文の構成は以下のようになっている. 第 2 章では本研究で用いた原理の説明を記し, 第 3 章において既存研究の問題定義を行い, 独自の提案手法を提案, また実際の実験方法を記す. そして第 4 章では第 3 章を踏まえての実験結果を記す. 第 5 章においては, 本研究の結果を踏まえ考察を記し, 最後の 6 章において結論を示す.

第 2 章　関連技術

・既存研究[2]

本研究の元となっている既存研究においては図 2.1 のフローチャートの様に判別を行っていた．具体的には，単語の悪口度を用いて悪口文/非悪口文の文分類を行っており，“単語の悪口度”を求めるために SO-PMI(Se-mantic Orientation Using Pointwise Mutual Information)を使用し，Web の検索結果などから共起度を算出することで単語自体が悪口かどうかを判別し，最終的には係り受け解析を行い悪性の文章かどうかを判別する．

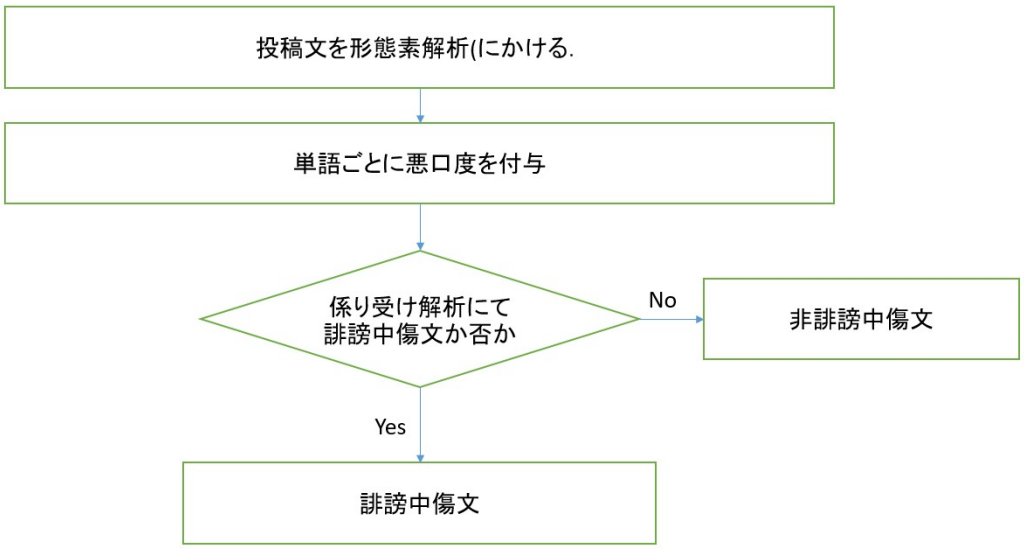

図 2.1 既存研究における誹謗中傷文の判別方法

・MeCab

MeCab とは，奈良先端科学技術大学院大学出身の Google エンジニア，工藤拓氏によって作られた，自然言語処理分野で多用されるオープンソースの形態素解析エンジン[3]である．元々は“ChaSen”という形態素解析ツールを基に作られていたが，今では独立して一から開発されている．

・MeCab-Ipadic-Neologd

MeCab-Ipadic-Neologd とは，形態素解析エンジン MeCab とともに使う単語分かち書き辞書のことを言う．週 2回程度の辞書更新により，新語・固有表現に強く，語彙数が多いという特徴がある．本研究においては，新語・造語への対応が課題となっていたので，形態素解析の分かち書き辞書として本辞書を使用した．

・glob

glob とは，ファイル一覧取得用のモジュール(モジュール:import 文を用いて呼び出すことで使える.py 形式のファイル)のことである.“*.txt”などと記すことで，その階層におけるテキストファイルをすべて取得することが出来る.

・NumPy

NumPy とは Python において数値計算を効率的に行うための拡張ライブラリであり，効率的に行うために型付きの多次元配列のサポートなどを行う.

・Pandas

Pandas とは Python においてデータ分析を効率的に行うためのライブラリである. このライブラリを用いることでデータの読み込みや統計量の表示，グラフ化，データ分析にかんする作業が容易に行えるようになる.

・Word2vec

Word2vec とは，大量のテキストデータを解析し，単語ごとの意味をベクトル表現化することにより，ベクトル計算によって単語同士の意味の近似度合いを数値化したりすることが出来るツールである． 例えば下の図 2.2 の例では単語間の関係性をベクトル演算することで，「王様−男+女＝女王様」となる. [4]

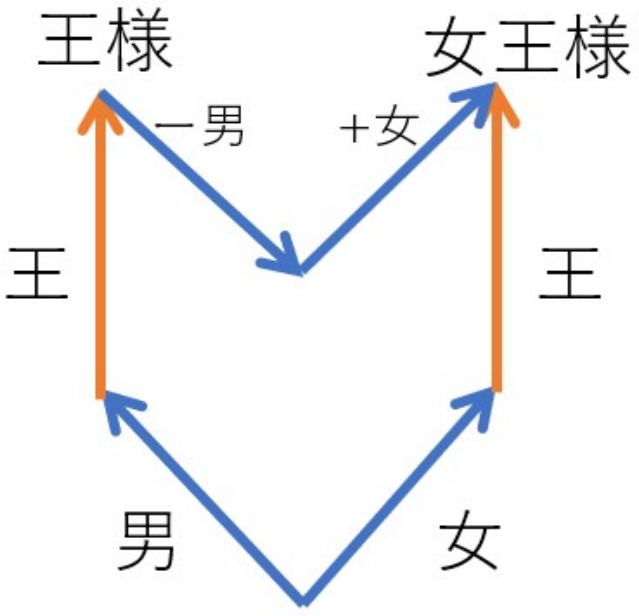

図 2.2 Word2vec における単語間のベクトル演算の例

・Tweet Archiver

Google スプレッド上において使用できるアドオンの一種. このアドオンを用いることで特定のワードを含むツイートを新しいものからすべて自動で取得することが出来る. ・

・gensim

大量の文章からトピックを分類し, 与えられた文章がどのトピックに属するかを分類知りたいときに使うライブラリである. gensim を使うことにより複雑な文章分類も手軽に行うことが出来るため, 自然言語処理においてはよく使われるライブラリである.

・Google Colaboratory

Google Colaboratory[5]とは, ブラウザから Python を記述, 実行できるサービスであり, 環境構築が不要で, クラウド上で GPU を使った演算などが行える点から学生のみならずデータサイエンティストや AI リサーチャーまでたくさんの人が使用するブラウザ上の開発環境のことである.

・scikit-learn (sklearn)

Scikit-learn は Python の機械学習ライブラリであり, 様々な機械学習のアルゴリズムが同じような書き方により, 実装できる点にある. また, はじめからサンプルのデータセットが付属されているため, インストールをしてすぐに機械学習を試すことが出来る特徴がある.

第 3 章　提案手法

本章においては既存研究の問題定義を行い，独自の提案手法を提案，また実際の実験方法を記す．

3.1 提案手法

まず 1 章で記した SNS における誹謗中傷の問題を解決するために，誹謗中傷文をより正確に判別するシステムについて新規手法を提案する．既存研究[2]においては,巨大掲示板をコーパスとし SO-PMI を用いて単語ごとに悪性であるかどうかを判別し，最後に文全体を悪性のものかどうか係り受け解析を用いて判別していたが，その手法であると新語・造語に対応できない点や SNS のような短文の個人ブログ特有の言い回しなどに対応できないとして，課題点が残っていた．

その現状を解決するために，まずデータセットを巨大掲示板から Twitter に変更し，既存研究では形態素解析を用いる際に MeCab とその付属の辞書をそのまま使用していたところを，MeCab Ipadic Neologd に変更する．そして SO-PMI を用いて単語ごとに悪口かどうかを判別していたが，本研究においては文全体をニューラルネットワークを用いて判断することで新語，造語自体が悪性かどうかわからない場合でも，関連語や，文一文を学習させることで悪性かどうかを判断できると考えた．その例として「笑」の有無による属性識別の可能性が例として挙げられ，(図 3.1)既存研究では，単語ごとに悪性度合いを見ているため，語尾に 1 文字～3 文字程度「笑」が存在していても形態素解析の際にそもそもデータとして考慮されていない場合や，悪性度合いをプラスマイナスする要素として考慮されていなかった．しかしながら本提案手法においては「笑」の有無によって中の単語の悪性度合いが分からなくても，良性か悪性かどうかを考慮する情報が増えることで，より正確な判別ができるのではないかと考えたため手法を提案した．

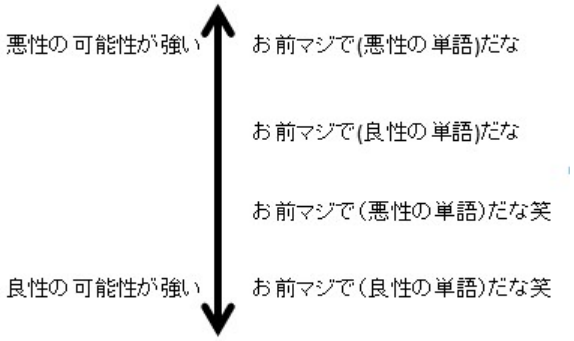

図 3.1 「笑」の有無による属性判別の可能性

そして上記のシステムを用いて誹謗中傷文を判別し，誹謗中傷文の可能性が高い場合，電話番号や生体情報などといった個人情報を求め，誹謗中傷文の可能性が低い場合は，個人情報の提出が匿名性の確保ができるSNSのシステムの提案を行う．

本提案において投稿を完全に拒否するのではなく敢えて個人情報を取得する方向性にした理由として，本研究において判別精度が 100%ではないので投稿自体を拒否してしまうことにより SNS 本来の表現の自由が奪われてしまう恐れがあると考えたため個人情報，生体情報の開示を求め，個人の特定を一早く，確実に行い，投稿者の本人の指紋等で誰が投稿したかを完全に特定する方法を取った．

3.2 実験方法

本研究の実験の対象として誹謗中傷文を判別するシステムの実験を行う．そして実験方法として Google Colaboratory 上にプログラムを構築し，実装を行った．具体的には，良性データ，悪性データを収集し訓練データとテストデータに分割後，実際に学習をさせてテスト用データを用いて評価を行った．データセットの取得方法，機械学習の実装方法については以下に記す．

（データセット取得）ファイル名が“topic-tweet-[ID].txt”と名付けられたデータセット(テキストファイル)を 120 個(良性 100 個(ID5903225～5903325),悪性 20 個(ID6003570～6003590))用意した．それぞれのデータセットの例を以下に示す．

（例 1） 私はあなたを愛するために生まれてきた.(良性データ)[6]
（例 2）マジでメンヘラは死んだらいい.　　（悪性データ）

（データセット取得：良性）良性のデータセットの取得方法として，Tweet をトレンドランキング等から自分自身が明らかに良性だと判断したものをランダムに取得し，手動で空白，絵文字，改行を取り除いたテキストデータを，自動的にテキストを入力することでテキストファイルを自動生成するプログラムに入力しテキスト形式で保存を行った．(図 3.1，図 3.2)

```
Set Sampledata:
今日は彼女とデートに行く日!!
Set Sampledata:
今日彼女に会ったんだけど，ふられたからガチでしんどい.
Set Sampledata:
```

図 3.1 データセットプログラム

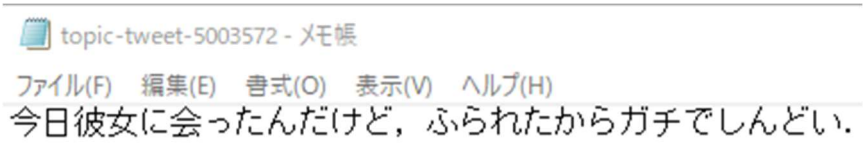

図 3.2 データセットのテキストデータ

（データセット取得:悪性）悪性のデータセットの取得方法として，Tweet データをグーグルスプレッド上において，アドオンの一種である Tweet Archiver を用いて「死ね」「うざい」のワードを含む日本語のツイートのみを自動で収集し，(図 3.3,図 3.4)自分自身が明らかに悪性であると判断したものを手動で空白，絵文字，改行を取り除いたテキストデータを，良性データの保存の際にも用いたテキストファイルを自動生成するプログラムに入力しテキスト形式で保存を行った．

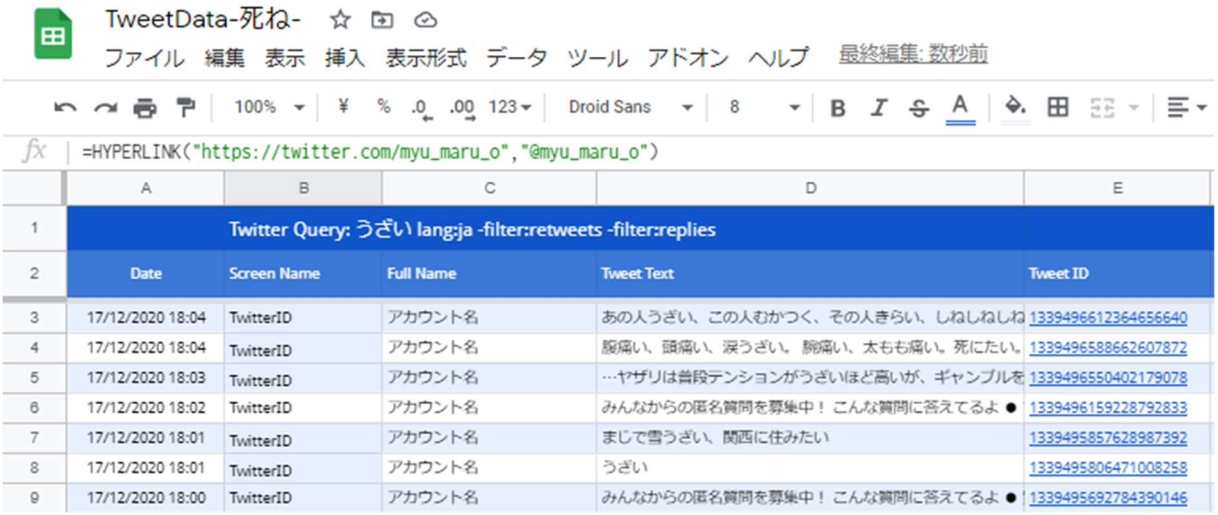

図 3.3 Tweet Archiver を用いて収集を行った Tweet データ

（データセット取得:ラベルセット）最後に Excel 上にラベルデータと Tweet データのテキストファイル ID の一覧のテーブルを作成し labeldata.csv(図 3.4)を作成した．テーブル一覧を作成する理由としてプログラム内において訓練データ，テストデータを分割する際に用いたり，評価の際にテストデータの答え合わせに用いたりするためにこのテーブルを作成した．

表 3.1 ID と label データテーブル

	ID	label
0	5903225	良性
1	5903226	良性
2	5903227	良性
3	5903228	良性
4	5903229	良性
...	...	...
115	6003586	悪性
116	6003587	悪性
117	6003588	悪性
118	6003589	悪性
119	6003590	悪性

（工程 1)まず日本語版,matplotlib,MeCab,MeCab Ipadic Neologd,MeCab-python3 のインストールを行った. そして次に, 以下の表のインポートを行った.

表 3.2 インポートしたライブラリ一覧

ライブラリ名	用途
glob	ファイル一覧用のライブラリ
numpy	数値計算用ライブラリ
pandas	データ分析用のライブラリ
matplotlib	可視化ライブラリ
japaneze_matplotlib	日本語化matplotライブラリ
seaborn(日本語設定済)	データ可視化ライブラリ
MeCab	形態素解析ライブラリ
gensim	Word2vecの類似度計算用ライブラリ
sklearn.model_selection ->train_test_split	訓練データとテスト用データ分割のライブラリ
sklearn.metrics ->accuracy_score	精度計算のライブラリ

そして以上に記したライブラリをインポート後, Wikipedia で学習済みの Word2vec モデルの取得を行った. 以上の準備が終わった後に, まず初めに MeCab[3] の標準辞書から MeCabIpadicNeologd への変更を行い, 引数からテキストを取得して形態素解析

と単語ごとに分割し配列に格納するクラス(devide_words())を作成した. 作成後そのクラスが正しく動くかテストを行った.

(工程 2)今回は事前に Wikipedia の日本語記事において学習した Word2vec の強力なモデル Wikipedia Entity Vectors(jawiki.word_vectors.300d.txt.bz2)[7]を用いてメモリ上に読み込んだ. そしてこちらでも正しく動作するか確認するためのプログラムを記述した.

(工程 3)次に準備段階で用意した label と ID の CSV ファイルを pd.read_csv を用いて読み込ませ, カテゴリデータとして配列 category_data に格納した. それと同時に, GooleDrive を Google Colaboratory にマウントした.

(工程 4)判別対象とする文章を形態素解析, 単語ごとの分かち書き, さらに配列 Words に格納して Word2vec に配列を渡し, 平均のベクトルを算出するクラスである(calc_future_vec(text))を作成した.

(工程 5)glob モジュールを用いて GoogleDrive 上のデータセット(テキストファイル)を参照し, calc_future_vec(text)を用いてデータセットの文章一つ一つに平均のベクトルを算出しリスト averaged_vec に格納した.

(工程 6)category_data に格納していたカテゴリデータから“label”を抽出し, リスト labels に格納した. そして, テキストのデータセットと label データを訓練データ(data_train,data_test,label_train,label_test)とテストデータに分割した. テストデータサイズは 0.1(10%=20 個)とし, 残りを訓練データとした.

(工程 7)Sklearn のサポートベクタマシン(SVM)を読み込み, オブジェクト生成を行った. そして実際に訓練を実施させてテストデータを予測し, 予測に関して実際の正解ラベルと比較し精度を出力した.

第 4 章　実験結果

まず, devide_words()が正しく動くかどうかのテストを行ったが, このテスト結果について下の図に示す.

```
devide_words("彼女はペンパイナッポーアッポーペンと恋ダンスを踊った。")
```

```
['彼女', 'は', 'ペンパイナッポーアッポーペン', 'と', '恋ダンス', 'を', '踊っ', 'た', '。']
```

図　4.1　devide_words() 動作確認結果

devide_words()の動作確認結果より, 単語ごとに分割されて正しく動作されていることが分かる.そして MeCab-Ipadic-Neologd を用いていることで新しい言葉(辞書で定義されておらず, 数年以内に生まれた言葉)などにも対応しており, 分かち書き辞書の設定においても正しく行えていることが分かる.

最後に学習後のテストデータを使った予測に関しての正解ラベルとの精度の差であるが, こちらも以下の図のような結果となった.

```
# テストデータを使った予測に関して、実際の正解ラベルと比較し精度を出力する
print (accuracy_score(label_test, label_predict))
```

```
0.9166666666666666
```

図　4.2 学習後のテスト結果

学習後の正解ラベルとの一致率は 91%という結果が出た.

そして最後に calc_future_vec(text)に手動でテキストを打ち込み, その文が良性か悪性かどちらに判別されるかどうかも実験したのでその結果を以下の図に示す.

```
feature_vec = calc_feature_vec('死んでください. ')
estimator.predict([feature_vec])
```

```
array(['良性'], dtype='<U2')
```

図　4.3 手動で打ち込んだ文章の判別結果(悪性)

```
feature_vec = calc_feature_vec('今日もめっちゃいい天気！')
estimator.predict([feature_vec])
```

```
array(['良性'], dtype='<U2')
```

図 4.4 手動で打ち込んだ文章の判別結果(良性)

結果としては，悪性の文章（“死ね”，“うざい”を含む文章）であっても良性と出てしまう結果となった．

第 5 章　考察

5.1 誹謗中傷文の判別研究における考察

まず研究結果から得られたデータをもとに考察を行う.

本研究において, 誹謗中傷文の判別訓練データを用いた学習後のテストデータを用いたテスト結果から精度 91%という結果が出たが, 実際に文章を入力して正しく判別できた悪性データの識別率は 50%程度であった.

その原因として偏りのあるデータセットにより, 正しく識別ができなかったと考えられる. しかしながら, 偏りはあるものの「死ね」,「うざい」等の言葉を含む文の一部は悪性と判断されるため, 学習自体は成功しておりデータセットの改良により結果は改善すると考える. しっかり学習, 評価を行うことが出来た点や devide_words()の動作確認においては正しく動作しているので成功したと考えられ, プログラム上にも問題はないと考えられる.

そして学習後のテストデータを用いた予測に関しての実際の正解ラベルとの比較精度については, 91%と高い数字が出たが手動で打ち込んだ文章の判別結果はそのパーセンテージに見合う結果ではなかった. なぜそのような結果が生まれたか考えた結果, 可能性としてあげられるのは各属性のデータセット数の割合である. 良性データは今回 100 個なのに対し, 悪性データは 20 個しか取り入れていなかった. それにより, テスト時においても各属性から 10%抽出した上でのテストとなるので, 良性データから 10 個, 悪性データから 2 個がテストデータとして抽出される計算になる.

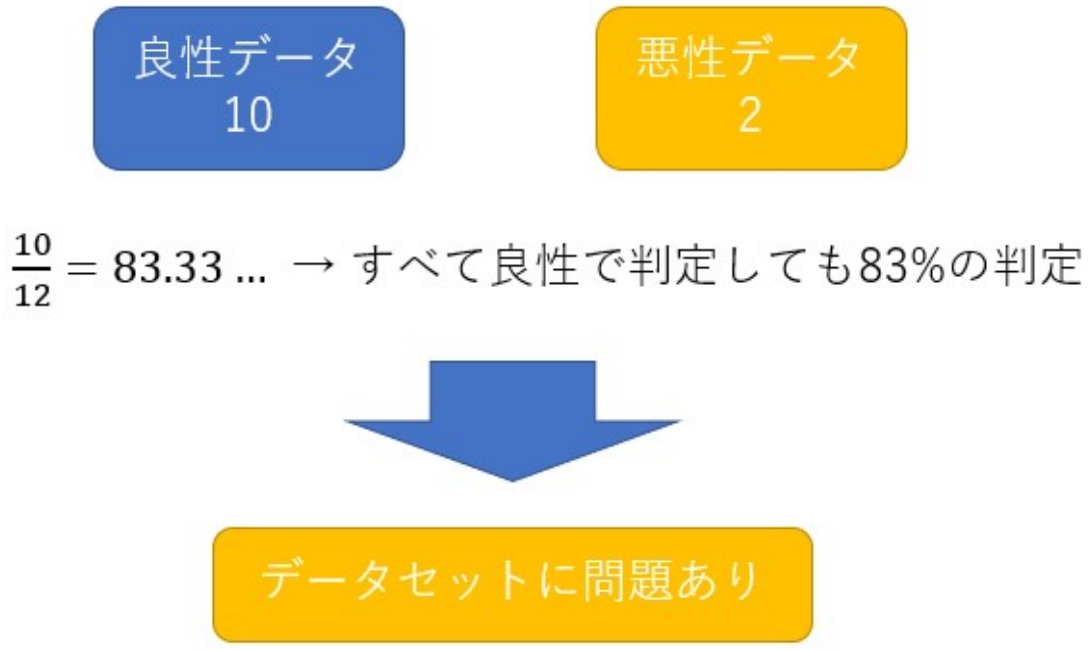

図 5.1 テスト結果における考察

それを踏まえると，図 5.1 の様に精度 91.666..%というのは 12 個のデータに対して 11 個のデータが正解であった計算になるが，ほとんどのデータが良性であるため全部を“良性”と判定したとしても 83.333..%の割合は出力される計算になる．よって悪性かどうかを見抜けられる実質的な割合は 50%程度になると考えられる．また今回悪性データを集める際に以下のように「死ね」，「うざい」を中心にスプレッドシート上に集めてデータセット化したため，悪性のデータとして上記の2ワードが存在することが必須であるかのような偏りのある学習方法になったのではないかと考え，もう一度改めてデータセットの取得方法について検討する必要があると感じた．

仲良くなると死ねとか暴言めっちゃ吐きます😉👌	1333638186229121026
フルブジオン軍勢最弱だから死ね	1333638175147786240
勉強しろとは言わないけれど、死ねばいいのに	1333638174455709696
西山死ね死ねしね	1333638143560409088
基本的にガチャにお金入れてる時は疲れてるから死ねって冗談	1333638074601934848
なんでそんなに要領悪いの？バカなのは私だけじゃなかったの	1333638047657705475
シーフ：武道に「死ね」って言われた。「それが人にものを頼	1333638031593488385
カイル「アルファベットって昔は２５文字だったって知ってる	1333638021116096512
だまれ おめぇも一緒だよ。 気持ち悪い。 同じように首だけ突	1333638020797382657
主が死ねば、縛られた魂は解放されるが、一度呪いを受け入れ	1333638017756459009
電電2年後期 量子力学I→ミリもわからん 電磁気学II→本来なら	1333637982645936129

図 5.2 「死ね」を含む Tweet の取得結果の一部

5.2 誹謗中傷の判別システムを有する SNS における今後の課題

次に本研究のテーマであった新たな SNS 上での誹謗中傷防止策として以下のポイントを提案する．

(1) 誹謗中傷文の可能性がある場合個人情報を提供してもらい本人確認を実施する．（図 5.2）
(2) 一つの投稿の返信欄において投稿順に返信を表示にするのではなく，ランダム表示にさせる．
(3) 画像，音声の投稿を禁止にする．
(4) 個人ごとに誹謗中傷ワード等の設定を行い，そのデータを用いて個人ごとにコミットした誹謗中傷文の判定を行う．

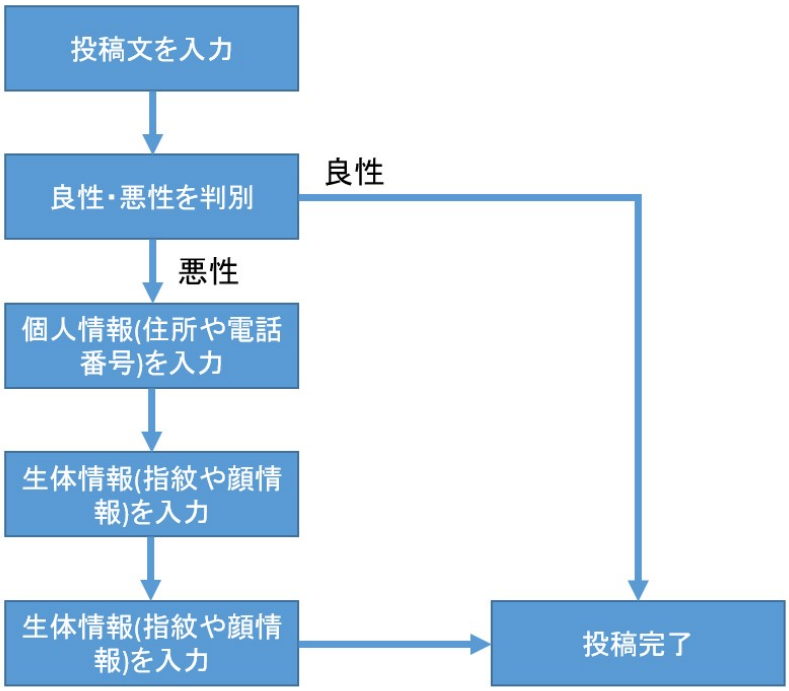

図 5.2 投稿完了までの一連の流れ

(1)におけるポイントは誹謗中傷文の可能性がある場合，個人情報を提供してもらうことで個人の特定が容易になる点である．現在のシステムでは，プロバイダに開示請求を行い ＩＰアドレスから住所の参照を行うことができるが，2,3 カ月の月日がかかってしまう問題点があった．しかし，本人確認をおこなうことで数日で本人確認が可能となる．そして迅速に法的手続きなどを行うことができる．

(2)におけるポイントは下記のように投稿順で並べることで一文字ずつをつなげて誹謗中傷文を完成させるのを防ぐためにこの機能の実装が必須だと考える．この機能により意図的に一文字ずつ並べることで完成する誹謗中傷文

への対策はできると考える.

図 5.3 一文字ずつ投稿することで完成する誹謗中傷文の例

(3) 本システムには画像，音声の良性・悪性を求める機能がないためそれらの実装ができるまでは投稿を禁止にする必要があると考える.

(4)個人によって誹謗中傷と感じる言葉に違いが生まれる. 例えば「ゴリラ」であったり，個人のコンプレックス等から生まれる誹謗中傷文が存在する事実がある. そのような言葉に対して，個人で設定を行いNGワードリストに追加し，それを含めた文章において誹謗中傷文の可能性を高くするシステムの導入が必要だと考える.

本対策においては，個人情報等の情報開示によって，投稿が許される形になるが判別精度が 100%でない為，完全に投稿を禁止することは誤判別をした際にＳＮＳ本来の機能が失われることを考え，あえて投稿できるようにする必要があると考える.

第 6 章　あとがき

本研究においてまえがきに挙げた「誹謗中傷文を判別し本人確認を行う新たなSNS」の誹謗中傷文判別手法を本研究において提案した. そして誹謗中傷文の判別実験結果から精度 91%という結果が出たが, 実際には悪性データの識別率は 50%程度で偏りのあるデータセットにより, 正しく識別ができなかった. しかしながら, 偏りはあるものの「死ね」,「うざい」等の言葉を含む文の一部は悪性と判断されるため, 学習自体は成功しておりデータセットの改良により結果は改善すると考える. そしてこのように問題点を明確にできたことで今後の課題点として明らかになった点については本研究においてよかった部分ではないかと考える.

そして本研究においてもう一つ「本人確認を行う新たな SNS」の構想を提案していたが, 誹謗中傷文の判別精度が 100%であれば投稿文自体を投稿不可にすれば 良いだけの話ではあるが, 現時点の技術ではそこまでの正確な判別は難しい. よって匿名性を維持しつつ, 疑わしい投稿には SNS の表現の自由の維持しながら個人情報を記録しておく方法がベストとだと考えた. このような現状の SNS 課題点として挙げられていた点を 5.2 節で述べたポイントを踏まえて対策案を提案できたので本研究においては成功であるといえる.

以上を踏まえて, 本研究は失敗, 課題点が残る結果となったが今後の必要な対策が明確化されたことで今後の研究の糧となる研究はできたのではないかと考える.

謝辞

本論文を執筆するにあたり，テーマ決めから研究の進め方までご指導を仰いでいただきました，岡村真吾先生には大変感謝しております．そして自然言語処理の内容について基礎の基礎から教えていただいた市川嘉裕先生，そして市川研究室のメンバーにも刺激的な議論をいただき助かりました．ここに感謝いたします．

参考文献

[1] 総務省:“SNS 上での誹謗中傷への対策に関する取り組みの大枠について”
URL:https://www.soumu.go.jp/main_content/000695577.pdf
閲覧日 2021 年 1 月 14 日

[2] 石坂達也, 山本和英:“Web 上の誹謗中傷を表す文の自動検出” 言語処理学会第 17 年次大会 発表論文集

[3] 形態素解析システム MeCab
URL: MeCab - 日本語形態素解析システム (dendai.ac.jp)
閲覧日 2021 年 1 月 14 日

[4] “Word2vec とは”
URL: https://ledge.ai/word2vec/
閲覧日 2021 年 2 月 24 日

[5] “Google Colaboratory”
URL: https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja
閲覧日 2021 年 1 月 21 日

[6]Queen:“Twitter”
URL:https://twitter.com/MeikakuGen/status/1300117303384657926
閲覧日 2020 年 12 月 5 日

[7] StevenBird,EwanKlein,Edward Loper:“入門自然言語処理”O’REILLY(2010)