

学 士 論 文

心療カウンセリングにおける会
題 目 話データのカテゴリ分類に関する研究

指導教員 小山田 耕二 教授

京都大学工学部 電気電子工学科

氏 名 林田祐磨

平成 29 年 2 月 10 日

目次

第 1 章	序論	1
第 2 章	関連研究	3
第 3 章	分類手法	6
3.1	分類手法の概要	6
3.2	word2vec を用いた単語のベクトル化	6
3.3	文のベクトル化	8
3.4	文ベクトルを入力とする機械学習	9
第 4 章	実験内容, 結果と考察	12
4.1	実験概要	12
4.2	実験結果	13
4.2.1	実験内容	13
4.2.2	実験結果	14
4.3	考察	16
第 5 章	結論と今後の課題	20
5.1	結論	20
5.2	今後の課題	20
	謝 辞	21
	参 考 文 献	22

第1章 序論

心療において、ストレス等に依る心の悩みを持ったクライアントに対してカウンセラーはカウンセリングを行う。その中でカウンセラーがクライアントの問題意識、つまりどのような「対人関係上の問題」を感じているかに注意して会話を引き出すことがカウンセリングの基本である¹⁾。しかし新人カウンセラーは熟練カウンセラーに比べて、自らの質問に対するクライアントの回答として、「対人関係上の問題」を引き出す事に関して未熟であるという問題がある。クライアントから「対人関係上の問題」を引き出すためには、カウンセラーの質問内容が重要とされている。

そこで新人カウンセラーに対し、熟練カウンセラーが指導を行うスーパービジョンの機会が設けられている。スーパービジョンの事例として、「心療内科における摂食障害専門ヨガ療法グループ」事例検討会²⁾では新人カウンセラーとクライアントのカウンセリング内容を動画で撮影し、その動画から書き起こされたテキストデータを熟練カウンセラーが読んで議論を行っている。その中で、新人カウンセラーは、自分の関心でカウンセリングを進めてしまいがちで、自分の中で作り上げた解釈内容をクライアントに確認するための「閉じられた質問」を多用する傾向が顕著であるとされる。しかし熟練カウンセラーがカウンセリング内容の文字を読むだけで、これらの会話の流れを十分に理解することは困難とされている。そのため、カウンセラーのどのような質問に対して、クライアントからどのような回答が得られたかについて、視覚的に理解できる仕組みが求められていた。

上辻らはこのような要求に対して、カウンセリングにおける会話の流れの可視化システムを開発した。それによって、このシステムではカウンセリングの会話の流れが時間軸に沿って可視化され、カウンセラーのどのような質問で、クライアントのどのような回答を引き出せたかについて、視覚的に理解できることを示した³⁾。

会話の流れの可視化システムの中でカウンセリングテキストデータとして取り扱うヨガ療法では、アドラー心理学が取り入れられている。アドラー心理学では、「人生のすべての問題は3つの主要な課題に分類することができる。つまり、交友の課題、仕事の課題、愛の課題である。」と唱えた⁴⁾。以上より、ヨガ療法ではクライアントの課題を、クライアントにとっての親疎の関係から、持続しない人間関係についての「仕事の課題」、持続するが、運命を共にしない人間関係についての「交友

の課題」、永続し運命を共にする関係についての「愛の課題」、と3つに区別している。また、クライアントが発言した話題がどの課題に関するものか、さらにカウンセリングの会話のやりとりの中でその課題に関する発言がどのように推移しているかを分析することは、カウンセリングプロセスを明確にするために役立つ。したがって、会話の流れの可視化システムでは、クライアントの会話内容を上で述べた3つの課題について、カテゴリ分類を行っている。

会話の流れの可視化システムではクライアントの発言内容を1発言単位ではなく1文単位で「愛」、「交友」、「仕事」に分類している。ここで1文とは、句点やクエスチョンマークで区切られた単位である。また、会話の流れ可視化システムにおけるクライアントの発言内容の分類手法を述べると、まずクライアントの発言を書き起こしたテキストデータの各文に対して、形態素解析を行い、文を各単語に分ける。次に、各カテゴリに含まれると思われる単語をカテゴリ毎に指定しておき、クライアントの発言文を構成する単語と一致する単語の数が一番多いカテゴリに分類し、カテゴリ毎に指定された単語と1つも一致しなかった文は「未分類」とする。

しかし上に述べた分類手法では、分類結果が予め指定した単語に大きく依存するといった問題がある。例えば、「友達」と「友だち」のような表記ゆれや、「夫」と「旦那」などの同義語が、同じ意味の単語であると認識できないといったことである。さらに、「夫の仕事がいつも遅いのが原因であまり眠れません。」という文は、「愛」のカテゴリに分類されるべきであるが、「仕事」という単語が「仕事」カテゴリの辞書に登録されているために誤って分類されてしまうという問題もある。このような誤った分類結果に関しては、ユーザーである熟練カウンセラーがシステム上で分類結果を確認し、誤った分類がなされている箇所を手動修正する必要がある、作業の負担がかかる問題が指摘されている。そのため、カウンセリングにおけるクライアントの発言内容をより正確に自動分類することが求められている。

本研究では、機械学習を用いてクライアントの発言内容に対するカテゴリ分類を行い、会話の流れの可視化システムにおける分類手法と比べて高い分類精度で自動カテゴリ分類を行うことができるか検証を行う。本論文の構成は次の通りである。第1章は本論文の序論である。第2章では、本論文の関連研究を挙げる。第3章では、本研究における、クライアントの発言内容の分類手法について述べ、第4章では本研究における分類結果を述べ、本研究における分類結果と会話の流れ可視化システムにおける分類結果との比較に対しての考察について述べる。第5章では本論文の結論と、本研究の今後の課題について述べる。

第2章 関連研究

本章では，本研究との関連研究を示し，その位置づけについて述べる．

テキスト分類は，スパムメールの自動振り分けやニュース記事の自動分類など様々な目的で行われている．近年，機械学習を用いたテキスト分類に関する研究や，その中でも教師付き学習によるテキスト分類に関する研究が数多く行われている．平ら⁵⁾は機械学習によるテキスト分類問題に対して，出現頻度の小さい単語まで考慮した学習を行わなければ分類精度が落ちることを述べ，高次元の単語ベクトルを用いるために SVM を用いた学習を行うことで，ニュース記事の分類において高い分類精度を実現した．この研究ではニュース記事の各文書の中から名詞を抽出し，Bag-of-words モデルによって文書をベクトル化している．ここで Bag-of-words とは文書中の単語の並びなどは考えず，文書に単語が含まれているかどうかのみを考えるモデルである．

従来ではニュース記事のように，ある程度文章が長く単語数や単語の種類も多い文書のベクトル化の際には Bag-of-words モデルがよく使われてきたが，本研究での分類対象はクライアントの発言 1 文毎であり，文の長さが短く単語数も少ないものが多い．そのため名詞を抽出し Bag-of-words モデルを用いて各文をベクトル化しても，その文の特徴が表れにくいと考えられる．

短い文の分類では，Sriram ら⁶⁾は Twitter の Tweet 内容を「ニュース」や「イベント」など 5 つの目的別に自動カテゴリ分類する手法を提案した．この研究では Bag-of-words による素性に加えて，Tweet の中に略語やスラングが使われているか，時間や場所についての記載があるか，など 8 つの特徴を基にした素性も加えることで，Bag-of-words のみによる素性を用いるよりも高い精度で Tweet の自動カテゴリ分類を行った．Sriram らの研究での分類カテゴリは，例えば「イベント」であれば時間や場所についての記載が多い，などのカテゴリ毎の Tweet の特徴が顕著であるが，本研究での分類カテゴリである「愛」「仕事」「交友」にはそのような顕著な特徴は存在しない．Bag-of-words モデルのデメリットとして，「友達」と「友だち」などの表記ゆれや，「父親」と「父さん」のような同義語を，全く別の単語として捉えるといった点が挙げられる．さらに，文書のベクトル次元数が学習する全コーパス中の語彙数と等しくなるため，本研究でも学習コーパスの語彙数は約 2 万でありベクトル次元

数も2万にも及ぶ。そのためニューラルネットワークを用いて学習を行うと、計算時間が膨大になる問題がある Bag-of-words によるベクトルを次元圧縮したものを用いる手法も考えられるが、永田ら⁷⁾によって次元圧縮により分類精度が下がったことが示されている。

Mikolov ら⁸⁾ は単語の分散表現を学習して単語のベクトル化を行う word2vec を提案した。word2vec では数百次元程度の密なベクトルで単語を高い精度で表現することが可能であり、現在もその用途について様々な研究が行われている。word2vec により得られた単語の分散表現を用いて、単語間の意味的な類似度を求めることが可能であり、日本語の研究として単語の意味を取り扱う研究が行われている。野沢ら⁹⁾ は、大量のレシピデータから食材と調理法を抽出し、word2vec で学習させ、word2vec で得られた単語ベクトルから各単語に類似する単語を算出し代替食材を発見する手法を提案した。また、菅原ら¹⁰⁾ は単語の分散表現を用いて多義語の語義曖昧性を解消する手法を提案した。語義曖昧性というのは、例えば”cool”という単語は「涼しい」や「かっこいい」など複数の語義を持つために、文脈により語義が異なることを言う。そこで word2vec により得た単語の分散表現を用いて、文書中における多義語の最もふさわしい語義を選ぶことを目的としている。しかし、word2vec により得た単語ベクトルを基に文章ベクトルを作成し、機械学習による文章分類に応用している日本語の研究事例は少ない。

日本語以外を取り扱う、word2vec により得た単語ベクトルを基にした文書分類では、Xing ら¹¹⁾ は、word2vec で得た単語ベクトルと、LDA モデルを用いた単語ベクトルを用いてそれぞれで文書ベクトルを作成した後に機械学習を行い、中国語のニュース記事の自動分類精度を比較した。機械学習アルゴリズムとしてはナイーブベイズ、k 近傍法、SVM を用い、その結果 word2vec で得た単語ベクトルを基に文書ベクトルを作成し SVM で機械学習を行う分類手法が最も精度が高かったことを示した。

加藤ら¹²⁾ は商品に対するレビューデータと評点に対し、word2vec と深層学習を用いて評判分析を行い、1-of-K ベクトルを用いたロジスティック回帰の性能とほぼ同程度であることを示した。しかし先に述べたように、本研究ではクライアントの発言1文毎を分類対象としているため、1-of-K ベクトルを用いての分類は不適当と考える。

本研究における手法として、word2vec により単語の分散表現を学習して得た単語ベクトルを基に、知恵袋の悩み相談に関する質問文を1文毎にベクトル化し、SVM

とニューラルネットワークによる機械学習を行い，クライアントの発言 1 文毎に対する自動カテゴリ分類を行った．その後，本研究における分類手法と会話の流れ可視化システムにおける分類手法との比較を行った．

第3章 分類手法

本章では、クライアントの発言内容のカテゴリ分類の手法について詳しく述べる．なお本研究では、会話の流れの可視化システムと同様にクライアントの発言内容を1文毎に分類している．クライアントの1文とは、句点かクエスチョンマークで区切られた単位のことである．また、クライアントの発言内容は上でも述べたように「愛」「交友」「仕事」という3つのカテゴリに分類する．

3.1 分類手法の概要

本研究における分類手法の概要を図3.1に示す．まず、Yahoo!知恵袋¹³⁾の悩み相談に関する文章をコーパスとして、word2vecを用いて、単語のベクトルを得る．word2vecにより得た単語のベクトルを基に、3つのどのカテゴリに属しているかのラベルを持ったYahoo!知恵袋の悩み相談に関する文のベクトルを教師付きデータとする．同様の手順で、クライアントの発言内容を書き起こしたテキストデータを1文毎にベクトル化し、これに正解カテゴリを付与したものをテストデータとする．教師付きデータを機械学習の入力データとし、学習させて、テストデータを学習したモデルに入力し、出力された予測カテゴリと予め与えられている正解ラベルとが一致するか調べ評価を行う．

3.2 word2vecを用いた単語のベクトル化

本研究で扱うデータは様々な単語から構成される文章の集まった文書データである．しかしコンピュータで単語や文を扱うために、単語や文を数値ベクトルとして表現する必要がある．機械学習により自動でテキストデータの分類を行う研究はこれまでに様々行われてきたが、多くの機械学習のアルゴリズムでは入力の次元数を学習前に定める必要がある．扱うテキストデータは単語の数、文や文章の長さがそれぞれ異なるものがほとんどであり、入力するテキストデータを固定長のベクトルで表現する必要がある．このような課題に対してこれまでに主に用いられている手法はbag-of-wordsである．bag-of-wordsは文章を単語の集合として捉え、単語の並び

方などは考慮せずに、単語の出現の有無と出現回数のみを考慮する手法である。しかしこの手法を用いると、学習に用いるテキストデータに含まれる語彙数だけベクトルの次元数が増えてしまい、語彙数と次元数が等しくなる。そのためニューラルネットワークなどのネットワークを用いた機械学習での入力次元が数万～数十万程度の次元になり、学習を行う際に計算時間が膨大になる。このような次元数の大きいベクトルに対して特異値分解による次元圧縮を行い、それを用いた分類⁶⁾も行われているが、数千次元まで圧縮すると分類精度が落ちることが報告されている。そこで、単語の分散表現を学習することで単語を数百次元程度の低次元のベクトルで表現する手法である word2vec が Mikolov ら⁷⁾により提案された。bag-of-words のようにベクトルの 1 つの次元のみが値を持ち、その他の次元の値が 0 となる疎なベクトル表現とは違い、分散表現では全ての次元が値を持っている。本研究においても機械学習を行う際に扱うテキストデータには数万以上の語彙数が含まれているため word2vec を用いて単語のベクトル化を行った。

word2vec に学習させるコーパスとしては Wikipedia やニュース記事のような大規模データが用いられることが多い。そこから得られた単語の分散表現を用いて機械学習などを用いて分類を行う研究では、分類の対象となるのがニュース記事などであることが多いためである。しかし本研究で扱うテキストデータはカウンセリングデータであり会話文である。Wikipedia やニュース記事は、固有名詞が比較的多く出現するが、個人の感情を表す形容詞はあまり含まれていない。しかしカウンセリングデータは会話文であるため固有名詞は少なく、感情を表す形容詞は比較的多く含まれる。しかし、カウンセリングの内容を書き起こしたテキストデータはあまり多く存在しておらず、且つプライベートな内容であるためインターネットなどでは公開されておらず、収集するのが困難である。そこで、カウンセリングデータとより似た内容の文章をコーパスに用いることを考えた。そのためカウンセリングデータに比較的内容が近いと思われる Yahoo!知恵袋に投稿されたものの中で悩み相談に関するカテゴリから文章を取得した。具体的には、本研究で分類を行う「愛」「交友」「仕事」の 3 カテゴリと内容が類似していると思われる、「恋愛相談」「家族関係の悩み」「友人関係の悩み」「職場の悩み」の 4 カテゴリから取得した。4 カテゴリの中から、1 つの質問とそれに対してのベストアンサーを 1 件として、各カテゴリ 12000 件、計 48000 件を取得してコーパスとした。また、word2vec は元々英語の文章を対象に考えられているため、コーパスは単語同士が半角スペースで区切られている必要がある。そのため、本研究では形態素解析ツール MeCab¹⁴⁾ を用いて予め知恵袋の文章

を単語毎に区切った文章を入力した。また、コーパスに対して形態素解析を行う際に、全ての単語を基本形で出力している。

次に word2vec で学習を行う際のパラメータについて述べる。ベクトルの次元数に関して、word2vec の開発者である Mikolov らによると、コーパスのデータサイズが増えるにつれてベクトルの次元数も大きくすべきとされている¹⁵⁾。また、データサイズがあまり大きくないにも関わらずベクトルの次元数を大きくしすぎると、精度が落ちることも報告されている。デフォルトの次元数は 100 次元であり、本研究で用いたコーパスはデータサイズがおおよそ 40MB、語彙数がおおよそ 1 万語と大きくないため、ベクトルの次元数は 200 以下とした。窓長は各窓長に対して学習後の各単語の意味的な類似度を調べた際に適切な値とされる 5 に設定した。学習アルゴリズムとして cbow モデルか Skip-gram モデルを選択できるが、本研究では Skip-gram モデルを選択した。また、min-count の値は 5 とし、コーパス内に 5 回未満しか出現しない単語は考慮しないとした。このような手順で単語の分散表現を得た。

3.3 文のベクトル化

次に 3.2 節で述べた単語の分散表現を用いて、文をベクトルに変換する。word2vec により単語の分散表現を得た際と同様にして、まず文を単語毎に区切る必要がある。そして文中に出現する各単語のベクトルを word2vec により得た分散表現から見つけ出し、それを用いて文ベクトル化を行う。具体例として、「母ともよく喧嘩しますし。」という文ベクトル化の流れを図 3.2 に示す。

1. 「母ともよく喧嘩しますし。」という文に対して形態素解析により単語毎の分かち書きを行い、基本形で出力する。
2. 文中に出現する全ての単語に対して、各単語のベクトルを word2vec により学習したモデルから取り出す。ただし文末の句点とクエスチョンマークは除いた。
3. 「母」、「とも」、「よく」、「喧嘩」、「する」、「ます」、「し」の単語ベクトルの 1 次元目の値を全て足し合わせ、各単語のベクトルの 1 次元目の和を計算する。このとき、もし単語が word2vec により学習したモデルに存在せずベクトルが存在しなかった場合、その単語は無視する。
4. 1 次元目に対して行った手順をベクトルの次元全てに対して行い、全ての次元分の和を計算する

5. 各文の長さは異なり、また単語の数も異なるので、足し合わせた単語数で各次元の和を割り、これを文のベクトルとする

つまり、各単語と文ベクトルの次元数を M として、文ベクトルを

$$d = [d_1, d_2, d_3, \dots, d_M] \quad (3.1)$$

とし、足し合わせた単語の、 i 番目の単語の j 次元目の要素の値を w_{ij} とすると、

$$d_1 = \frac{1}{N} \sum_{i=1}^n w_{i1}, \quad d_2 = \frac{1}{N} \sum_{i=1}^n w_{i2}, \quad \dots, \quad d_M = \frac{1}{N} \sum_{i=1}^n w_{iM} \quad (3.2)$$

と表せる。Xing ら¹¹⁾ は、機械学習による文書分類を行う際に、word2vec で得た単語ベクトルの単純な和の平均として文章ベクトルを作成する手法により、分類精度が高くなることを示している。したがって本研究でも、文ベクトルを単語ベクトルの和の平均として算出した。

3.4 文ベクトルを入力とする機械学習

本節では、本研究における学習と分類手法について述べる。本研究の目的は、予め指定された3つの「愛」「交友」「仕事」カテゴリに、どのカテゴリに属するかが未知である文を正確に自動カテゴリ分類することである。本研究で行う学習は教師付き学習であり、教師付き学習とは、入力に対して出力が指定されて学習を行う方法である。本研究では、3.3 節で述べた手順で算出した1文毎の文ベクトルを、入力である1文毎の特徴量として用いて、出力には「愛」「交友」「仕事」の3つの分類カテゴリを用いる。学習を行ったパターン認識器に対して、どのカテゴリに属するか未知であるクライアントの発言1文毎の文ベクトルを入力し、予測カテゴリを出力させることで、自動カテゴリ分類を行う。本研究では、機械学習に用いるアルゴリズムにSVM¹⁶⁾とニューラルネットワーク¹⁷⁾を用いた。SVMとはパターン認識モデルの1つであり、線形しきい素子を用いて2つのカテゴリのパターン識別機を構成する手法である。2つのカテゴリに分類を行う際に、ハードマージンSVMでは入力データが完全に線形分離可能であると仮定して、1つの誤分類も許容せずに分離超平面を決定する。つまり、 M 個の m 次元教師付きデータ $x_i (i=1, 2, \dots, M)$ がクラス1, 2いずれかに属するとして、ラベルをクラス1のときに $y_i=1$, クラス2の

ときに-1 とする．線形分離可能である場合，決定関数は w を m 次元係数ベクトル， b はバイアス項として

$$D(x) = w^T x + b \quad (3.3)$$

と決めることが出来る．また線形分離の条件から決定関数は

$$y_i(w^T x_i + b) \geq 1 (i = 1, \dots, M) \quad (3.4)$$

の条件を満たす．条件式 (3.4) を満たすような w , b を求め，入力データに対して分離超平面を決定するが，このとき分離超平面とそれにもっとも近い教師データとの間の距離をマージンとよぶ．条件式 (3.4) を満たす分離超平面は無数に存在するが，SVM ではマージンが最大となる超平面を識別境界とする．したがって，SVM では汎化能力が高いというのが大きな特徴である．しかし，本研究では入力データは完全に線形分離可能ではないと考え，ソフトマージン SVM を用いた．ソフトマージン SVM では線形分離可能でない場合に適用できるように， $\xi_i (\geq 0)$ を導入し条件式 (3.4) を拡張し，

$$y_i(w^T x_i + b) \geq 1 - \xi_i, (i = 1, \dots, M) \quad (3.5)$$

を満たし，

$$Q(w, b, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^M \xi_i \quad (3.6)$$

を最小にするような最適化問題を考え分離超平面を決定する．したがってハードマージン SVM とは異なり，ソフトマージン SVM では誤分類を許容する．ここで C は式 (3.6) の右辺第1項のマージン最大化と第2項の誤分類の最小化のトレードオフを決定するパラメータであり，分離超平面の決定に大きな影響を与えるため，適切に設定する必要がある．そして，このように入力空間内で非線形分離を行う必要がある場合には入力空間を高次元の特徴空間に写像して特徴空間上でマージンが最大となるように超平面を決定する方式がよく用いられており，カーネル法とよばれているが，本研究でもこの方式を用いた．カーネル法では，非負のラグランジュ乗数 α_i, β_i を導入して，決定関数は

$$D(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x_j) + b \quad (3.7)$$

となる．ここで $K(x_i, x_j)$ はカーネルであり，問題に応じてこのカーネルを適切に選ぶことで汎化能力を向上することができる．本研究では，

$$K(x, x') = x^T x' \quad (3.8)$$

で表される線形カーネル (linear) と,

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (3.9)$$

で表されるラジアル基底関数カーネル (rbf) を用いた. ここで γ は分布の半径を制御するパラメータであり, rbf カーネルにおいて超平面の決定に大きな影響を与える. また, 上でも述べた通り SVM は 2 つのカテゴリに分類する 2 値分類を基本としているが, 本研究では 3 つのカテゴリに分類することを目的としている. SVM において多カテゴリへの分類を行う場合, One-Vs-One 方式と One-Vs-The-Rest 方式のいずれかを用いる. 今, K カテゴリに分類することを考える. One-Vs-One 方式はある特定のカテゴリに属するか, また別の特定のカテゴリに属するかの 2 カテゴリ分類問題を解く分類器を, 全てのカテゴリの組み合わせ, つまり $K(K-1)/2$ 個使用する. One-Vs-The-Rest 方式はある特定のカテゴリに属するか, 他の $K-1$ 個のいずれかのカテゴリに属するかの 2 カテゴリ分類問題を解く分類器を, 全ての組み合わせ, K 個利用する.

ニューラルネットワークもパターン認識に用いられるモデルの 1 つである. 本研究で用いたニューラルネットワークは入力層と出力層を 1 つずつ持ち, 中間層を複数持つ多層ニューラルネットワークである. また全てのノード同士がリンクによって繋がれた, 全連結型のネットワークである. ニューラルネットワークの仕組みの概要として, 入力層の各ノードに値を入力し, それぞれの値が重みを持ったノード間のリンクを通して, 閾値を持った中間層の各ノードに受け渡され, その値が中間層で変換される. 中間層で変換される際には活性化関数を用いて変換が行われる. そして変換された値が同じように次の中間層へと受け渡されていき, 最後出力層に伝達し値を出力させる. そして出力した値が期待される値に近づくように, 各ノード間のリンクの重みやノードの閾値を最適化するように学習を行う. 本研究では入力層のノード数を文ベクトルの次元数と等しく固定し, ベクトルの各要素の値を入力層の各ノードに入力する. 出力層のノード数は分類カテゴリ数と同じ 3 で固定し, 出力が最も大きいカテゴリをその文が属するカテゴリであると判断する. また中間層の層数, ノード数は可変なパラメータとしており, このパラメータによって精度が変わるため, SVM と同様に後述の実験により最適なパラメータを求める.

本研究での分類の手順は, 初めに 3.3 節で作成した教師データを上で述べた機械学習の入力として学習を行う. そして学習後のモデルにテストデータを入力し, 「愛」「交友」「仕事」のいずれかの予測カテゴリを出力させ 3 カテゴリへの分類を行う.

第4章 実験内容，結果と考察

本章では，本研究における分類手法の分類精度を高めるための実験，会話の流れの可視化システムにおける分類手法との比較を行う．さらに，実験結果に対する考察を述べる．そして本研究における分類手法の問題点を述べる．

4.1 実験概要

本研究で機械学習を行うために使用したライブラリについて述べる．SVMによる学習を行うために，オープンソースの機械学習ライブラリである scikit-learn を使用する．次にニューラルネットワークによる学習を行うために，Google が開発しオープンソースとして公開されている Tensorflow を使用する．共に使用したプログラム言語は Python である．次に本研究の実験で使用したテキストデータについて以下に述べる．word2vec の学習に用いたコーパスは，Yahoo!知恵袋の「恋愛相談」，「友人関係の悩み」，「家族関係の悩み」，「職場の悩み」4 カテゴリーの，各カテゴリ 6000 件ずつ，質問 24000 件，質問に対するベストアンサー 24000 件の計 48000 件である．機械学習の教師データに用いたテキストデータは，Yahoo!知恵袋の「家族関係の悩み」，「友人関係の悩み」，「職場の悩み」カテゴリの質問文を 1 文毎に区切った，各カテゴリ 6000 文ずつ，計 18000 文である．機械学習のテストデータに用いたテキストデータは，クライアントの発言のうち，熟練カウンセラーの手動修正後「愛」「交友」「仕事」いずれかのカテゴリに分類されていた 170 文である．word2vec 学習に用いたテキストデータの 1 件とは，投稿者 1 人の投稿文章である．word2vec に用いたコーパスは単語数 6298390，語彙数 18100，サイズ 37.4MB である．教師データに関して，質問文のみを用いるのは，本研究における分類対象はクライアントの発言内容であり，相談することを目的として Yahoo!知恵袋に投稿した投稿者の質問文が内容的に比較的類似していると考えたためである．そして質問文を 1 文毎に区切り，3.3 節で述べた手順に従って，1 文毎に文ベクトルを作成する．ここで 1 文とは句点やクエスチョンマークで区切られた単位のことである．「家族関係の悩み」カテゴリから取得した質問文 1 文毎の文ベクトルに「愛」のラベル，「友人関係の悩み」カテゴリの質問文 1 文毎の文ベクトルに「交友」ラベル，「職場の悩み」カテゴリの

質問文 1 文毎の文ベクトルに「仕事」ラベルをそれぞれ付けて、それを教師データとする。テストデータに関して、クライアントの発言 1 文毎のうち、ユーザーの手動修正後「愛」「交友」「仕事」いずれかのカテゴリに分類されていた 170 文をテストデータに用いる。クライアントの発言 170 文に対して 3.3 節で述べた手順に従い文ベクトルを算出し、テストデータとした。第 1 章でも述べたように、会話の流れの可視化システムにおけるクライアントの発言の分類結果を、熟練カウンセラーが確認を行い、誤った分類箇所を手動修正している。その手動修正後の分類カテゴリをテストデータの正解カテゴリとする。しかし、クライアントの発言 1 文毎の 170 文の中には、カウンセラーからの質問に対する単なる返事などといった、その 1 文だけをカテゴリ分類するのが難しいと考えられる文も含まれている。そこで、

1. 「そのことで夜眠れなくて…」のように、「そのこと」が何のことなのかを把握する為には前の発言文に遡らなければならない文

2. 「不安です。」のように、カウンセラーからの質問に対する返事などの短い文

を 170 文から除外した 48 文に対しての実験も同様に行う。以下の実験を行う際に、分類精度を評価する指標として本研究では正答率を用いる。正答率とは、全てのテストデータに対して、正しいカテゴリを出力した割合であり、テストデータのうち正解カテゴリと同じ出力をした数を T 、正解カテゴリと異なる出力した数を F 、正答率 (%) を Accuracy とすると、

$$Accuracy = \frac{T}{T + F} \times 100 \quad (4.1)$$

と表すことができる。これらのテキストデータ、評価指標を用いて以下に述べる実験を行う。

4.2 実験結果

4.2.1 実験内容

初めに、SVM とニューラルネットワークそれぞれについて、本研究における分類手法の精度を高めるために、パラメータを調整する実験を行う。3.4 節で述べたように、SVM ではカーネル関数とパラメータ C および γ の選択によって分離超平面がそれぞれ異なり分類精度も大きく異なる。ニューラルネットワークでも同様に扱うデータにより適切なパラメータが異なる。したがって SVM とニューラルネットワークにおいてそれぞれパラメータの調整を行い、それぞれ入力するベクトルの次元数

も変えて実験を行う。以下にパラメータ調整のための実験内容の詳細を述べる。まず、SVMとニューラルネットワークに共通するパラメータとして、入力するデータの次元数である、3.2節で述べた手順で word2vec により得る単語ベクトルの次元数を 30, 40, 50, 100, 200 と変化させた。次に、SVM においては、3.4 節で述べたが、本研究は2値分類ではなく多カテゴリ分類であり、本研究では One-Vs-The-Rest 方式を使用した。カーネル関数には、非線形のラジアル基底関数カーネル (rbf カーネル) と線形カーネル (linear カーネル) を用いた。パラメータ C に関して、式 (3.6) より、 C を大きくするほど、 ξ_i が小さくなるような分離超平面を決定しなければならない、つまり誤分類を許容できなくなる。本研究では C を 0.1, 1, 10, 100, 1000 と変化させた。rbf カーネルのパラメータ γ に関して、 γ が小さいほど単純な識別境界となり、大きいほど複雑な識別境界となる。カーネル関数が rbf カーネルの場合のみ、 γ を 0.0000001, 0.000001, 0.00001, 0.0001, 0.001 と変化させた。ニューラルネットワークにおいては、活性化関数はソフトマックス関数で固定した。入力層と出力層の間の中間層の数を 1 層, 2 層とした。また、各中間層のノード数を 64, 128, 256, 512 と変化させた。エポック数は学習を繰り返し行う回数であり、学習データに対する精度とテストデータに対する精度が共に高くなるように適切に選ぶ必要がある。本研究ではエポック数を 40, 80, 160, 320 と変化させた。また、本章の初めにも述べたとおり、クライアントの発言 1 文毎 170 文に対する実験と 48 文に対する実験を述べるが、170 文に対する正答率がより良かった SVM のみ 48 文に対する実験を行う。続いて、上で述べた分類手法の精度を高めるために調整したパラメータの中で最も正答率の高かったパラメータを用いて、正答率を分類精度の指標として、本研究と会話の流れの可視化システムとの分類手法の比較を行う。

また、本研究と会話の流れの可視化システムにおける分類手法それぞれに対して、正しく分類できている文と分類できていない文を調べて比較を行う。

4.2.2 実験結果

学習アルゴリズムとして、SVMとニューラルネットワークを用いたそれぞれで、前節で述べた各パラメータの組み合わせごとのカテゴリ分類の正答率を述べ、その中で最も正答率の高かったパラメータを用いて、本研究における分類手法と会話の流れの可視化システムにおける分類手法との比較を行う。SVM を用いて、170 文に対する実験結果を述べる。SVM を用いてベクトル次元数、カーネル関数、 C 、 γ を変えながら行った、クライアントの発言 1 文毎の 170 文に対する自動カテゴリ分類の

正答率を表 4.1～4.5 に示す。170 文に対して最も正答率が高かったのは、ベクトル次元数 30, rbf カーネル, $C=1$, $\gamma=0.00001$ の場合で 63.5%であった。次元数に関して、次元数が 30, 40, 50 の場合は正答率にそれほど大きな差はなかったが、次元数が 100, 200 になると次元数が低い場合に比べて正答率が下がった。カーネル関数に関して、どの次元数においても rbf カーネルの方が linear カーネルよりも高い正答率を示した。C に関して、 $\gamma=0.001$, 0.01 と比較的大きい場合は C が最も小さい 0.1 の場合に正答率が高くなることが多く、 $\gamma=0.00001$, 0.0001 とそれよりも小さい場合には $C=1$ の場合に正答率が高くなるが多かった。 γ に関して、C の値によらず $\gamma=0.00001$, 0.0001 の場合に正答率が高いことが多い。また、 $\gamma=0.0000001$, 0.01 の場合には正答率が低い場合が多い。 γ が 0.000001 以下になると、全ての次元数, C の値において、linear カーネルの正答率と等しくなった。次に、SVM を用いて、48 文に対する実験結果を述べる。クライアントの発言 1 文毎の 48 文に対する自動カテゴリ分類の正答率を表 4.6～4.10 に示す。48 文に対して最も正答率が高かったのは、ベクトル次元数 40, rbf カーネル, $C=1$, $\gamma=0.00001$ の場合で 81.3%であった。カーネル関数に関して、どの次元数においても rbf カーネルの方が linear カーネルよりも高い正答率を示した。C に関して、 $C=1000$ の場合は正答率が低いが多かった。 $\gamma=0.001$, 0.1 と比較的大きい場合と、0.000001 以下と比較的小さい場合は C を 1000 から小さくしていくとほとんどの場合に正答率が上がっていったが、 $\gamma=0.00001$, 0.0001 の場合には C を 0.1 まで小さくすると正答率が下がるが多かった。 γ に関して、C の値によらず $\gamma=0.00001$, 0.0001 の場合に正答率が高くなるが多かった。 γ が 0.000001 以下になると、全ての次元数, C の値において linear カーネルの正答率と等しくなった。ニューラルネットワークを用いて、170 文に対する実験結果を述べる。ニューラルネットワークを用いてベクトル次元数, 中間層の数, 中間層のノード数, エポック数を変えながら行った、クライアントの発言 1 文毎の 170 文に対する自動カテゴリ分類の正答率を表 4.11～4.15 に示す。ニューラルネットワークでは、ベクトルの次元数, 中間層の数, 中間層のノード数, エポック数による正答率の顕著な傾向は現れず、SVM と比べると正答率は極端に低くなった。次に、会話の流れの可視化システムにおける分類手法との比較についての実験結果を述べる。本研究における分類手法と会話の流れの可視化システムにおける分類手法との 170 文, 48 文に対する分類結果をそれぞれ表 4.16, 4.17 に示す。本研究における分類手法には、170 文に対して最も正答率の高かった、SVM で、カーネル関数を rbf カーネル、ベクトル次元数を 30, $C=1$, $\gamma=0.00001$ とし

たものと、48 文に対して最も正答率の高かった、SVM で、カーネル関数を rbf カーネル、ベクトル次元数を 40、 $C=1$ 、 $\gamma=0.00001$ としたものを用いる。ただし、会話の流れの可視化システムにおける分類手法では分類結果が未分類となっている文が 170 文中 104 文存在しているが、未分類は誤ったカテゴリに分類したとみなして正答率を算出している。170 文と 48 文に対して共に会話の流れの可視化システムにおける分類手法よりも本研究における分類手法の方が正答率は上回っている。

4.3 考察

4.2 節の実験結果に対する考察を述べる。

学習アルゴリズムである SVM とニューラルネットワークに関して、ニューラルネットワークを用いた分類手法の正答率が低い理由としては、ニューラルネットワークの学習が学習データに対して過学習を起こしていると考えられる。ニューラルネットワークにおいてどの次元数、パラメータにおいても、学習データに対しては正答率が 80% を超えているが、クライアントの発言に対しての正答率は平均して 10% ほどである。学習データに用いた Yahoo! 知恵袋の質問文は、投稿者がどのカテゴリにどのような文章を投稿するかは自由であり、「脳がスイーツすぎませんか」、「レポートで弾性係数を調査する内容の課題が出ました。」といった 3 つの分類カテゴリに全く関係のない文章が含まれていることが過学習を起こしている原因の 1 つであると考えられる。また、投稿 1 件全体では「家族関係に関する悩み」の内容のものであるが、1 文毎に区切ったために、「来年、仕事を 1 年休みます。」という文のように「仕事」カテゴリに分類されるべきであろう文が「愛」カテゴリの文として学習されてしまうことも過学習を起こしている原因の 1 つであると考えられる。

それに対して SVM の学習ではマージンを最大化するような分離超平面を決定することで、学習データに対して過学習しすぎることなく、分類カテゴリが未知のクライアントの発言に対しても高い正答率を実現できたと考えられる。

しかし SVM でも、rbf カーネルの場合に式 (9) において γ を大きくすると、 $x \cdot x'$ が大きくなれば急速に指数関数項の値は 0 に近づく。すなわちそのカテゴリの教師データの近くのデータしかそのカテゴリに分類されなくなる。さらに C も大きくすると誤分類を許容しなくなるため、今回 C と γ を共に大きくした場合に正答率が下がっていると考えられる。

次にそれぞれの分類手法で分類できている文とできていない文について述べる。「母

ともよく喧嘩しますし.」,「最近お仕事でお休みを頂いているので.」という文のように,「母」,「仕事」といったカテゴリ特有の単語が出現する文は,会話の流れの可視化システムと本研究における分類手法で共に正しく分類できている.

クライアントの配偶者についての会話の中で,ユーザーの手動修正後「愛」カテゴリに分類されている文に,「貯金とか将来のことを考えて行動していないのではないか」,「そんなことをして生活費がどうなるのか」,「あの一部屋に住めると思いません.」という文がある.しかし会話の流れの可視化システムにおける分類手法では,これらの文に出現する単語がシステムの辞書には登録されておらず,正しく分類できていないが,本研究における分類手法では正しく分類できている.本研究における分類手法で正しく分類できているのは,学習データの「愛」カテゴリの文の中に「夫の収入が少なく,貯金できない.」,「私は彼と生活費を折半しています.」,「妹と母親が隣の部屋で会話をしていました.」のように,「貯金」,「生活費」,「部屋」の単語を含む文が,他のカテゴリの学習データに比べて多いことが理由であると考えられる.

本研究における分類手法で正しく分類できていなかった文の中で特徴的な文について述べる.クライアントの配偶者についての会話の中で,ユーザーの手動修正後「愛」カテゴリに分類されている文に,「あの人の仕事を否定したくありません.」,「貯金もしていないし,あんな仕事なので,もし事故を起こしたら ……」という文がある.しかし,本研究における分類手法では,「仕事」カテゴリに分類されている.これは,「仕事」のラベルをつけた学習データの文の中に「仕事」という単語を含む文が頻出しており,文中の他の単語よりも「仕事」という単語のベクトルがその文ベクトルに大きく影響しているためと考える.「あの人の仕事を否定したくありません.」というような文を正しく分類するために,文の係り受け解析を行い,「あの人」の「仕事」であることから「愛」のカテゴリであると分類できるようにする必要があると考える.

同様にクライアントの配偶者についての会話の中で,ユーザーの手動修正後「愛」カテゴリに分類されている文として,「お客さんに料理を出しすぎです.」,「将来設計なんかできません.」という文がある.しかし,本研究における分類手法では,「仕事」カテゴリに分類されている.理由の1つは,誰に関する話題であるか把握するための単語が抜けていることである.また,学習データの中に「料理」という単語が出現する回数は「愛」カテゴリでは219回であり,「仕事」カテゴリでは65回である.しかし「客」と「料理」が共に出現する文は,「簡単に注文された料理を注文

してくれた客に間違わずに運べるものですか。」という文のように、アルバイトや仕事に関する文がほとんどであり、「仕事」カテゴリの学習データに多く存在する。次に、「将来設計」という単語は形態素解析を行うと、「将来」と「設計」に分かれてしまう。「将来」という単語は「愛」カテゴリの学習データの方に多く出現するが、「設計」という単語は「仕事」カテゴリの学習データの中の設計の仕事に関する文に多く出現し、また「将来の仕事が不安です。」という文も「仕事」カテゴリの学習データに多く存在するため、「将来設計なんかできません。」という文が「仕事」カテゴリに分類されていると考える。形態素解析を行う際に、MeCabに予め備わっている辞書に加えて、「将来設計」のように名詞が連続した複合名詞が登録されている辞書を用いる必要があると考える。

「そういうやりすぎるところが嫌いで.」、「あの人のこだわりだと思います。」という文は「愛」カテゴリに分類されるべきであるが、前者は主語が抜けており、後者は「あの人」が誰なのか特定できず、文中の他の単語もいずれかのカテゴリの学習データに多く存在せず、共にどのカテゴリに関する文なのか判別し難い。このような文を正しく分類するために、その前の会話に遡り主語を補うことができるかの検討をする必要があると考える。「友人が家族とのストレスを。」という文は「交友」カテゴリに分類されるべきであるが、「愛」カテゴリに頻出する「家族」と「交友」カテゴリに頻出する「友人」という単語が共に出現して、「家族」を含む文の方が学習データに多く存在しているため「愛」カテゴリに分類されていると考える。また、本研究における分類手法を用いての、48文に対する正答率は170文に対する正答率よりも高いが、80%程度に留まっている理由として、クライアントの前の発言文に遡らないとカテゴリの把握ができない文に対しては正答率が低いためと考える。

クライアントの前の発言に遡らないとカテゴリが把握できない文や、カウンセラーからの質問に遡らないとカテゴリが把握できない返事などの短い文を正しく分類するために、今後クライアントの発言1文毎の分類ではなく、カウンセラーの質問やクライアントの前の発言を含めた、会話のブロック毎の分類を検討する必要があると考える。

分類カテゴリに関して、本研究では「愛」「交友」「仕事」の3つを分類カテゴリとした。そのためこれらの3つの分類カテゴリのいずれにも当てはまらないクライアントの発言文は、ユーザーの手動修正後のカテゴリを「未分類」としている。今後これらの文に対しても分類を行う必要があると考える。これらの文はクライアント自身の症状に関する記述であるものが多い。そこで、アドラー心理学において、ク

クライアントの課題のグループ分けに関して、「愛」「交友」「仕事」の3つ以外に、第4の課題とされる「自己」、第5の課題とされる「スピリチュアル」がある。これらのカテゴリを分類カテゴリに加えることで、本研究ではテストデータとしなかったクライアントの発言文に対してもカテゴリ分類を行うことを検討する必要がある。

第5章 結論と今後の課題

5.1 結論

本研究では、word2vecによって得られた単語のベクトルから1文毎の文ベクトルを算出し、SVMを用いて機械学習を行い、クライアントの発言1文毎の自動カテゴリ分類を行い、会話の流れの可視化システムにおける辞書に基づいたカテゴリ分類と比較して、高い分類精度を実現した。

しかし、前章で述べたとおり、ユーザーの手動修正の負担がかからない分類精度には及んでおらず、分類精度をさらに高める必要がある。さらにもっと多くの未知のクライアントの発言文に対して分類精度を検証する必要がある。次節に今後の課題について述べる。

5.2 今後の課題

本研究で得られたことを踏まえて、今後検討すべき課題について述べる。

- 分類対象のその1文だけでは、カテゴリ分類を行うための情報が少ない文に対してカテゴリ分類を行うために、その1文の前後の会話文も含めた会話のブロック毎の分類の検討
- 複数のカテゴリの、そのカテゴリ特有の単語が共に出現する文を分類するために、係り受け解析によりカテゴリの判別を行う手法の検討
- テストデータが十分な数とは言えないため、さらに多くのクライアントの発言に対してのカテゴリ分類を行うこと
- 学習データ数による分類の正答率の推移の検討
- 形態素解析を行う際に、MeCabに予め備わっている辞書に加えて、複合名詞を登録した辞書を用いること
- 本研究でテストデータとして扱わなかったクライアントの発言文をカテゴリ分類するために、分類カテゴリに「自己」、「スピリチュアル」の追加

謝 辞

本研究を進めるにあたり, 有益な御指導, 御助言を頂きました京都大学学術情報メディアセンタービジュアライゼーション研究分野の小山田耕二教授, 江原康生特定准教授, 夏川浩明特定助教, 尾上洋介特定助教に深く感謝致します.

本研究を進めるにあたり, プログラミング技術を始め, 様々な御助言を頂き協力して下さった, 京都大学大学院人間・環境学研究科修士課程 2 年生の今井晨介氏, 京都大学大学院工学研究科修士課程 1 年生の上辻智也氏, 梅澤浩然氏をはじめとする院生の先輩の皆様にはご協力を賜りました. ここに深く御礼申し上げます.

最後に, 家族をはじめとする私の学生生活を支えてくださったすべての皆様へ心から感謝の意を表します.

参考文献

- 1) 野田俊作, 続アドラー心理学トークンセミナー勇気づけの家族コミュニケーション, (アニマ 2001, 1991).
- 2) 認定ヨーガ療法士会, <http://yogatherapy-hyogo.net/lecture.html>, (2015).
- 3) 上辻智也, 鎌田穰, 小山田耕二, カウンセリングにおける会話の流れの可視化に関するユーザー評価, 第 44 回可視化情報シンポジウム 2016 論文集, (2016).
- 4) R. R. Ansbacher H. L. Ansbacher, *THE INDIVIDUAL PSYCHOLOGY OF ALFRED ADLER*, (Haper Row Publishers Inc, New York, 1956).
- 5) 春野雅彦平博順, Support vector machine によるテキスト分類, 情報処理学会研究報告自然言語処理 (NL) , Vol. 1998, No. 99, (1998), pp. 173–180.
- 6) D. Engin S. Bharath, F. Dave, Short text classification in twitter to improve information filtering, *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, (ACM, New York, NY, USA, 2010), pp. 841–842.
- 7) 佐々木稔永田純平, 文書分類をタスクとした pylearn2 の maxout+dropout の利用, 言語処理学会第 21 回年次大会, (2015), pp. 900–903.
- 8) G. Corrado T. Mikolov, K. Chen, Efficient estimation of word representations in vector space, *CoRR*, Vol. abs/1301.3781, (2013).
- 9) 山本修平野沢健人, word2vec を用いた代替食材の発見手法の提案 (データ工学と食メディア), 電子情報通信学会技術研究報告. DE, データ工学, Vol. 114, No. 204, (2014), pp. 41–46.
- 10) 菅原拓夢, 笹野遼平, 高村大也, 奥村学, 単語の分散表現を用いた語義曖昧性解消, 言語処理学会発表論文集, Vol.21, (2015), pp. 648–651.

- 11) X. Zhang C. Xing, D. Wang, Document classification with distributions of word vectors, *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference*, ().
- 12) 加藤和平, 大島考範, 二宮崇, word2vec と深層学習を用いた大規模評判分析, 言語処理学会第 21 回年次大会発表論文集, (2015), pp. 525–528.
- 13) Yahoo!知恵袋, <http://chiebukuro.yahoo.co.jp/>, (2016).
- 14) Mecab, <http://taku910.github.io/mecab/>, (2013).
- 15) Z. Geoffrey M. Tomas, Y. S. Wen-tau, Linguistic regularities in continuous space word representations, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*, (Association for Computational Linguistics, 2013).
- 16) 阿部重夫, パターン認識のためのサポートベクトルマシン入門, (森北出版, 2011).
- 17) 熊沢逸夫, 学習とニューラルネットワーク, (森北出版, 1998).

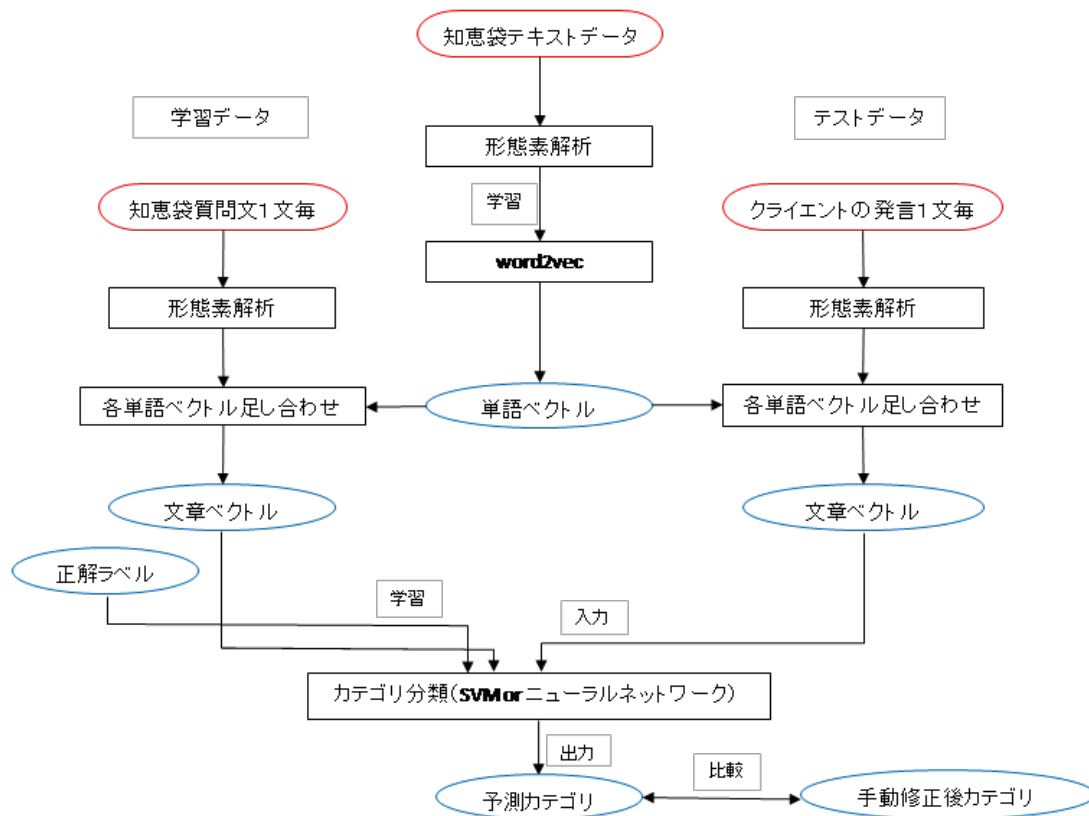


図 3.1: 自動カテゴリ分類のフローチャート図

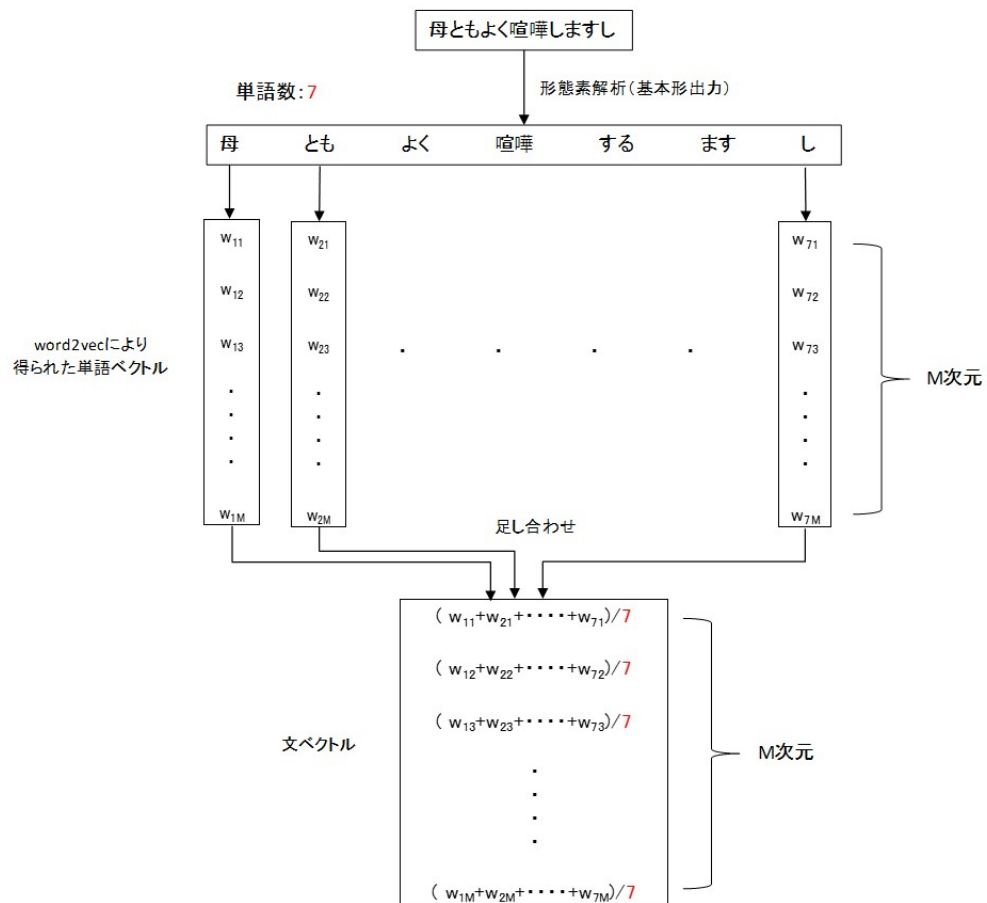


図 3.2: 文のベクトル化の流れ

表 4.1: SVM での次元数 30 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>		<i>rbf</i>				
				gamma				
				0.0000001	0.000001	0.00001	0.0001	0.001
C	0.1	40.0	40.0	40.0	48.8	56.5	42.4	40.6
	1	38.2	38.2	38.2	63.5	35.3	39.4	39.4
	10	38.2	38.2	38.2	39.4	37.6	39.4	38.2
	100	38.2	38.2	38.2	39.4	35.3	39.4	38.2
	1000	38.2	38.2	38.2	40.6	38.2	38.2	35.3

表 4.2: SVM での次元数 40 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>		<i>rbf</i>				
				gamma				
				0.0000001	0.000001	0.00001	0.0001	0.001
C	0.1	37.6	37.6	37.6	28.8	35.9	41.2	38.2
	1	37.6	37.6	37.6	60.6	37.6	38.2	38.2
	10	40.0	40.0	40.0	37.6	38.8	38.8	36.4
	100	40.0	40.0	40.0	40.0	38.8	36.5	38.8
	1000	40.0	40.0	40.0	38.8	36.5	38.2	38.8

表 4.3: SVM での次元数 50 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>		<i>rbf</i>				
				gamma				
				0.0000001	0.000001	0.00001	0.0001	0.001
C	0.1	39.4	39.4	39.4	52.4	35.3	40.6	39.4
	1	36.5	36.5	36.5	59.4	38.8	38.8	38.8
	10	37.6	37.6	37.6	43.5	39.4	38.8	38.8
	100	38.8	38.8	38.8	38.2	38.8	38.8	37.6
	1000	38.2	38.2	38.2	38.8	38.8	38.2	37.1

表 4.4: SVM での次元数 100 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>			<i>rbf</i>		
					gamma		
		0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	33.5	33.5	33.5	44.1	32.4	33.5
	1	32.9	32.9	32.9	42.9	32.4	34.7
	10	31.8	31.8	31.8	31.8	34.7	34.1
	100	30.6	30.6	30.6	35.9	34.1	34.7
	1000	30.6	30.6	30.6	34.7	34.7	31.8

表 4.5: SVM での次元数 200 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>			<i>rbf</i>		
					gamma		
		0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	32.4	32.4	32.4	31.8	35.3	28.8
	1	21.8	21.8	21.8	48.2	32.9	28.0
	10	21.8	21.8	21.8	39.4	31.0	32.3
	100	21.8	21.8	21.8	34.1	31.7	24.3
	1000	21.8	21.8	21.8	33.5	24.4	20.1

表 4.6: SVM での次元数 30 の 48 文に対する正答率 (%)

カーネル関数		<i>linear</i>			<i>rbf</i>		
					gamma		
		0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	56.3	56.3	56.3	68.8	77.1	58.3
	1	50.0	50.0	50.0	79.2	54.2	56.3
	10	50.0	50.0	50.0	58.3	56.3	54.2
	100	50.0	50.0	50.0	56.3	56.3	54.2
	1000	50.0	50.0	50.0	56.3	54.2	50.0

表 4.7: SVM での次元数 40 の 48 文に対する正答率 (%)

カーネル関数		<i>linear</i>			<i>rbf</i>		
					gamma		
			0.0000001	0.000001	0.00001	0.0001	0.001
C	0.1	56.3	56.3	56.3	33.3	45.8	60.4
	1	56.3	56.3	56.3	81.3	58.3	58.3
	10	56.3	56.3	56.3	56.3	58.3	58.3
	100	56.3	56.3	56.3	58.3	56.3	52.1
	1000	56.3	56.3	56.3	56.3	52.1	56.3

表 4.8: SVM での次元数 50 の 48 文に対する正答率 (%)

カーネル関数		<i>linear</i>			<i>rbf</i>		
					gamma		
			0.0000001	0.000001	0.00001	0.0001	0.001
C	0.1	58.3	58.3	58.3	72.9	41.7	58.3
	1	50.0	50.0	50.0	79.2	58.3	58.3
	10	54.2	54.2	54.2	64.6	58.3	58.3
	100	54.2	54.2	54.2	56.3	58.3	54.2
	1000	54.2	54.2	54.2	58.3	54.2	52.1

表 4.9: SVM での次元数 100 の 48 文に対する正答率 (%)

カーネル関数		<i>linear</i>			<i>rbf</i>		
					gamma		
			0.0000001	0.000001	0.00001	0.0001	0.001
C	0.1	43.8	43.8	43.8	58.3	39.6	50.0
	1	41.7	41.7	41.7	56.3	43.8	50.0
	10	41.7	41.7	41.7	43.8	52.1	52.1
	100	37.5	37.5	37.5	54.2	50.0	43.8
	1000	39.6	39.6	39.6	52.1	43.8	39.6

表 4.10: SVM での次元数 200 の 48 文に対する正答率 (%)

カーネル関数		<i>linear</i>	<i>rbf</i>					
			gamma					
			0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	56.3	56.3	56.3	41.7	60.4	50.0	58.3
	1	43.8	43.8	43.8	75.0	62.5	52.1	58.3
	10	39.6	39.6	39.6	66.7	58.3	58.3	43.8
	100	39.6	39.6	39.6	58.3	58.3	43.8	39.6
	1000	39.6	39.6	39.6	58.3	43.8	39.6	39.6

表 4.11: ニューラルネットワークでの次元数 30 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>	<i>rbf</i>					
			gamma					
			0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	40.0	40.0	40.0	48.8	56.5	42.4	40.6
	1	38.2	38.2	38.2	63.5	35.3	39.4	39.4
	10	38.2	38.2	38.2	39.4	37.6	39.4	38.2
	100	38.2	38.2	38.2	39.4	35.3	39.4	38.2
	1000	38.2	38.2	38.2	40.6	38.2	38.2	35.3

表 4.12: ニューラルネットワークでの次元数 40 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>	<i>rbf</i>					
			gamma					
			0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	37.6	37.6	37.6	28.8	35.9	41.2	38.2
	1	37.6	37.6	37.6	60.6	37.6	38.2	38.2
	10	40.0	40.0	40.0	37.6	38.8	38.8	36.4
	100	40.0	40.0	40.0	40.0	38.8	36.5	38.8
	1000	40.0	40.0	40.0	38.8	36.5	38.2	38.8

表 4.13: ニューラルネットワークでの次元数 50 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>		<i>rbf</i>			
				gamma			
		0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	39.4	39.4	39.4	52.4	35.3	40.6
	1	36.5	36.5	36.5	59.4	38.8	38.8
	10	37.6	37.6	37.6	43.5	39.4	38.8
	100	38.8	38.8	38.8	38.2	38.8	38.8
	1000	38.2	38.2	38.2	38.8	38.8	37.1

表 4.14: ニューラルネットワークでの次元数 100 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>		<i>rbf</i>			
				gamma			
		0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	33.5	33.5	33.5	44.1	32.4	33.5
	1	32.9	32.9	32.9	42.9	32.4	34.7
	10	31.8	31.8	31.8	31.8	34.7	34.1
	100	30.6	30.6	30.6	35.9	34.1	34.7
	1000	30.6	30.6	30.6	34.7	34.7	31.8

表 4.15: ニューラルネットワークでの次元数 200 の 170 文に対する正答率 (%)

カーネル関数		<i>linear</i>		<i>rbf</i>			
				gamma			
		0.0000001	0.000001	0.00001	0.0001	0.001	0.01
C	0.1	32.4	32.4	32.4	31.8	35.3	28.8
	1	21.8	21.8	21.8	48.2	32.9	28.0
	10	21.8	21.8	21.8	39.4	31.0	32.3
	100	21.8	21.8	21.8	34.1	31.7	24.3
	1000	21.8	21.8	21.8	33.5	24.4	20.1

表 4.16: 170 文に対する分類手法の比較

会話の流れの可視化システム		SVM, 次元数30 rbf, C=1, gamma=0.00001
正解した文の数	49	108
正答率(%)	28.8	63.5

表 4.17: 48 文に対する分類手法の比較

会話の流れの可視化システム		SVM, 次元数40 rbf, C=1, gamma=0.00001
正解した文の数	14	39
正答率(%)	29.2	81.3