修　士　学　位　論　文

複数特徴ベクトルの組み合わせによる感情分析の領域適応

平成 30 年度

茨城大学大学院理工学研究科

情報工学専攻

ZHAO　XINYU

平成 30 年度茨城大学大学院理工学研究科情報工学専攻　修士学位論文

複数特徴ベクトルの組み合わせによる感情分析の領域適応

氏名：17NM711A　ZHAO XINYU
指導教員：新納 浩幸 教授

論文要旨

本論文では感情分析の領域適応のタスクに対して、文書に対する複数特徴ベクトルを組み合わせて利用する手法を提案し、Amazon Dataset の領域 books, dvd, music の 3 つを利用して、計 6 通りの領域適応評価実験によって単一特徴ベクトルによる領域適応の比較を行い、その考察を述べる。

自然言語処理の多くのタスクにおいて、教師あり学習の手法が大きな成功を収めている。ただし実際の応用場面では学習に使ったデータのソース領域 S と、解析対象となるデータのターゲット領域 T が異なることが多い。この場合、領域 S で学習できた分類器が領域 T では有効に機能しない。例えば感情分析では “books” に関するレビューの positive/negative 判定を学習した分類器が“dvd”に関するレビューの positive/negative をうまく判定できない。これは Domain Shift と呼ばれる問題であり、この問題に対処するための領域適応の研究が近年活発に行われている領域適応には様々なタイプの問題が存在するため、その手法も多岐にわたるが、感情分析にタスクを限れば、領域 S のデータと領域 T のデータを shared feature subspace にマップする手法が有力である（Anders Søgaard, 2013）。

領域適応には様々なタイプの問題が存在するため、その手法も多岐にわたるが、感情分析にタスクを限れば、領域 S のデータと領域 T のデータを shared feature subspace W にマップする手法が有力である。空間 W 上で学習と識別を行えば Domain Shift の問題は起こらない。問題は W の構築方法である。代表的な研究は MMD（Borgwardt et al., 2006）であり、近年は深層学習を利用した手法も多数提案されている（Patel et al.,2015）。

本論文ではタスクを感情分析とする。この場合、データは文書なので W を文書の埋め込み表現の空間に設定する。また文書を低次元空間に埋め込む方法としてもいくつかの手法があるが、領域間の差異の多様性から様々な領域適応に対処できる単一の埋め込み手法は存在しない。そこでここでは文書の埋め込み手法を組み合わせて利用することを提案する。具体的には文書の埋め込み手法として doc2vec（Lau and Baldwin, 2016）と特異値分解を組み合わせる。文書 d を doc2vec と特異値分解を利用して、それぞれ埋め込み表現 e と g に変換する。d の bag-of-words のモデルによるベクトルを v としたとき、通常分類器の学習では特徴ベクトルとして v が用いられるが、本手法では v の代わりに、上記 3 つのベクトルを連結した $[v:e:g]$ を特徴ベクトルとして用いる。

実験では Amazon Dataset を利用した。Amazon Dataset の領域 books, dvd, music の 3 つを利用して、計 6 通りの Domain Shift を試す。各 Domain Shift で正解率を求め、それら平均値によって手法を評価する。特徴ベクトルとして v や $[v:e]$ あるいは¥([v : g] ¥)を用いた場合とを比較することで、本手法の有効性を示す。

Master's Thesis in Scholastic 2018, Major in Computer and Information Sciences,
Granduate School of Science and Engineering, Ibaraki University

Domain Adaptation Using a Combination of Multiple Feature Vectors for Sentiment Analysis

Author: ZHAO XINYU (17NM711A)
Adviser: Prof. Hiroyuki Shinnou

ABSTRACT

In this paper, we propose a new method for domain adaptation by using a combination of multiple embeddings for sentiment analysis. In the experiment, we used an Amazon dataset that has three domains ("books"', "DVD" and "music") and six types of domain adaptations. The experiment showed the effectiveness of our proposed method.

Supervised learning has achieved great success in many natural language processing tasks. However, in real applications, the source domain *S* in the learning stage may be different from the target domain *T* in the testing stage. In this case, the classifier learned in the source domain *S* is ineffective in the target domain *T*. For example, in sentiment analysis, the classifier learned through reviews on "books" was ineffective for judging whether a review on "DVD" is positive or negtive. This problem is known as the ``domain shift'' problem. In recent years, research on domain adaptation has been very active to solve this problem (Anders Søgaard, 2013).

There are various types of problems in domain adaptation. However, in the case of domain adaptation of sentiment analysis, the method that maps data in domains *S* and *T* to the shared feature subspace *W* is very effective. When we conduct learning and testing on the space *W* , a domain shift does not occur. Therefore, the manner in which to construct *W* becomes the problem. For this problem, the representative research is maximum mean discrepancies (MMD) (Borgwardt et al., 2006). In recent years, many methods using deep learning have been proposed (Patel et al.,2015).

This study focuses on the sentiment analysis task, wherein the data is the document and *W* is the space of embeddings for the document. Furthermore, there are several methods to embed documentts in the low-dimensional space, but there is no single embedding algorithm that can be used to respond to various domain adaptations from the diversity of domain differences. Therefore, we combine the embeddings of the document in this study, specifically the combination of doc2vec (Lau and Baldwin, 2016) and singular value decomposition (SVD). The document *d* is converted to the embedding *e* and *g* by using doc2vec and SVD respectively. Moreover we use the vector *v* obtained from the bag-of-words (BOW) model of the document *d*. Generally, *v* is used as the feature vector to learn the classifier. In this study, we use three connected vectors, namely [*v* : *e* : *g*] as feature vectors instead of *v* . In the experiment, we used an Amazon dataset that has three domains (" books"', "DVD" and "music"); then we designed six types of domain adaptations. Thereafter, the method was evaluated on the basis of the average accuracy of each domain adaptation. By comparing the use of *v*, [*v* : *e*] or [*v*: *g*] as feature vectors, the effectiveness of our proposed method is demonstrated.

目次

表目次

図目次

第 1 章

序論

1.1 概要

本論文では感情分析の領域適応のタスクに対して、文書に対する複数特徴ベクトルを組み合わせて利用する手法を提案し、Amazon Dataset の領域 books, dvd, music の 3 つを利用して、計 6 通りの領域適応評価実験によって単一特徴ベクトルによる領域適応の比較を行い、その考察を述べる。

自然言語処理の多くのタスクにおいて、教師あり学習の手法が大きな成功を収めている。ただし実際の応用場面では学習に使ったデータのソース領域 S と、解析対象となるデータのターゲット領域 T が異なることが多い。この場合、領域 S で学習できた分類器が領域 T では有効に機能しない。例えば感情分析では “books” に関するレビューの positive/negative 判定を学習した分類器が“dvd”に関するレビューの positive/negative をうまく判定できない。これは Domain Shift と呼ばれる問題であり、この問題に対処するための領域適応の研究が近年活発に行われている領域適応には様々なタイプの問題が存在するため、その手法も多岐にわたるが、感情分析にタスクを限れば、領域 S のデータと領域 T のデータを shared feature subspace にマップする手法が有力である（Anders Søgaard, 2013）。

領域適応には様々なタイプの問題が存在するため、その手法も多岐にわたるが、感情分析にタスクを限れば、領域 S のデータと領域 T のデータを shared feature subspace W にマップする手法が有力である。空間 W 上で学習と識別を行えば Domain Shift の問題は起こらない。問題は W の構築方法である。代表的な研究は MMD（Borgwardt et al., 2006）であり、近年は深層学習を利用した手法も多数提案されている（Patel et al.,2015）。

本論文ではタスクを感情分析とする。この場合、データは文書なので W を文書の埋め込み表現の空間に設定する。また文書を低次元空間に埋め込む方法としてもいくつかの手法があるが、領域間の差異の多様性から様々な領域適応に対処できる単一の埋め込み手法は存在しない。そこでここでは文書の埋め込み手法を組み合わせて利用することを提案する。具体的には文書の埋め込み手法として doc2vec（Lau and Baldwin, 2016）と特異値分解を組み合わせる。文書 d を doc2vec と特異値分解を利用して、それぞれ埋め込み表現 e と g に変換する。d の bag-of-words のモデルによるベクトルを v としたとき、通常分類器の学習では特徴ベクトルとして v が用いられるが、本手法では v の代わりに、上記 3 つのベクトルを連結した $[v : e : g]$ を特徴ベクトルとして用いる。

実験では Amazon Dataset を利用した。Amazon Dataset の領域 books, dvd, music の 3 つを利用して、計 6 通りの Domain Shift を試す。各 Domain Shift で正解率を求め、それら平均値によって手法を評価する。特徴ベクトルとして v や $[v : e]$ あるいは¥([v : g] ¥)を用いた場合とを比較することで、本手法の有効性を示す。

1.2 本論文の構成

本論文では、はじめに理論とその手法について紹介する。第 2 章では感情分析の領域適応の背景と関連研究を説明する。第 3 章では文書の埋め込み表現を構築する手法の doc2vec と SVD について述べる。第 4 章で複数特徴ベクトルを組み合わせて利用し実験の内容と結果を示す。第 5 章ではその実験結果について考察する。最後に第 6 章で結論を述べる。

第 2 章

感情分析の領域適応

領域適応は、ターゲット領域のラベル付きデータを用いる教師ありの手法と、それを用いない教師なしの手法に大別できる。教師ありの手法の場合、Daumé の手法 (Daumé III, Hal, 2007) が簡易でしかも能力が高いため、標準手法となっている。

本論文で扱うのは教師なしの手法である。教師なしの手法の場合、素性ベースのものと事例ベースのものに大別できる (Pan and Yang, 2010)。事例ベースの手法は学習データの事例に重みを付けた学習であり、共変量シフトを仮定する。共変量シフトとは$P_S(c|x) = P_T(c|x)$かつ$P_S(x) = P_T(x)$という仮定である。共変量シフト下では、ソース領域のデータ x に対して確率密度比$\mathrm{r} = P_T(x) / P_S(x)$を重みとした重み付き学習から、$P_S(c|x)$を得ることができる (Sugiyama and Kawanabe, 2011)。素性ベースの手法は学習データの素性に重みを付けた学習であり、古典的には SCL (Blitzer et al., 2006) が有名である。あるいは素性ベースの手法はソース領域のデータとターゲット領域のデータを shared feature subspace W にマップする手法とも見なせ、MMD はその代表的な研究である (Borgwardt et al., 2006)。またこのタイプの研究として CORAL (Sun et al., 2016) は簡易でしかも能力が高いため近年注目されている。また深層学習を用いた領域適応の手法も基本的に素性ベースの手法と見なせる(Glorot et al., 2011)。CORAL を拡張した手法(Sun and Saenko, 2016)や敵対性ネットワークを利用した手法(Ganin and Lempitsky, 2015)(Tzeng et al., 2017)などがある。

第 3 章

埋め込み表現による領域適応

タスクが感情分析の場合データは文書になるので、文書の埋め込み表現を求める手法により shared feature subspace が構築できる。そこで本論文では doc2vec により s−hared feature subspace を構築する。また文書の埋め込み表現を求める手法は、bag of words のモデルで表現した文書ベクトルを次元縮約する手法とも見なせる。そのため特異値分解を利用しても shared feature subspace が構築できる。

3.1 doc2vec

doc2vec とは文書の埋め込み表現を構築する手法である（Lau and Baldwin, 2016）。doc2vec を説明するためにはまず word2vec の説明が必要。word2vec は単語の埋め込み表現を構築する手法であり、文書をベクトル化して、単語同士の意味をある程度計算でいるようになる。その正体は隠れ層と出力層の 2 層のニューラルネットワークです。このニューラルネットワークに次々に単語を読み込ませて重みを学習させていくのですが、word2vec で獲得できる単語のベクトル表現というのはネットワークの重みそのものです。word2vec には CBOW，skip-gram の 2 つのアーキテクチャがある。

3.1.1 CBOW と skip-gram

CBOW モデルと skip-gram モデルのニューラルネットワークは図 3.1 と図 3.2 を表示する。

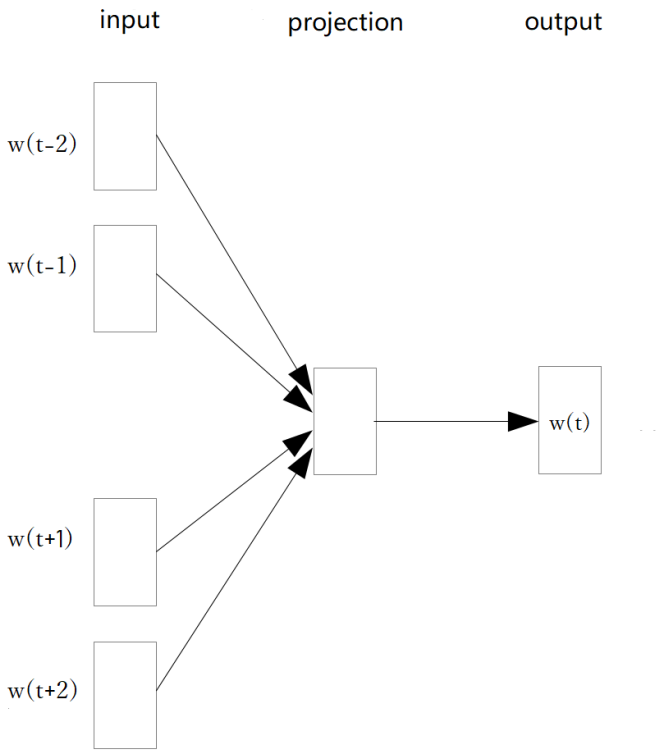

図 3.1　CBOW モデル

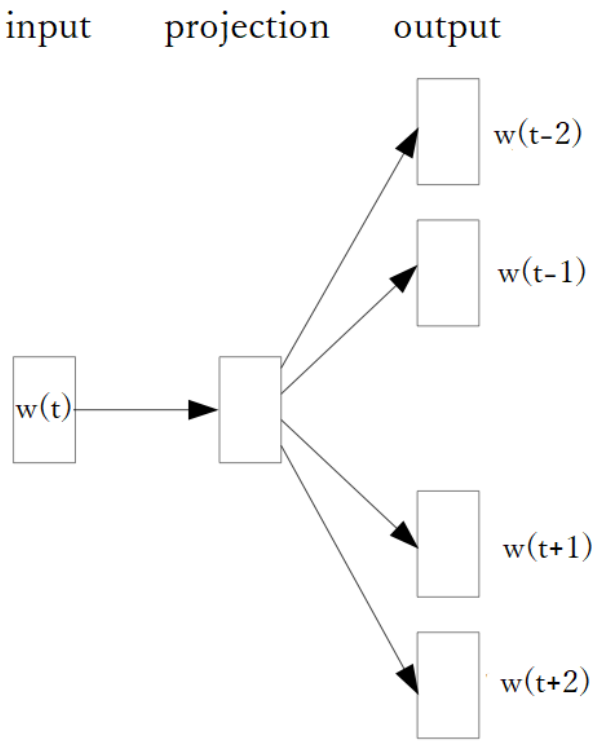

図 3.2　skip-gram モデル

CBOW (Continuous Bag-of-Words)は単語周りの言葉から指定の単語を推定するモデルです。目標単語前後の n 個単語を文脈として入力して，着目している単語 W(t) を推定します。

skip−gram 簡単と言えばある単語を与えた時にその周辺語を予測するためのモデルです。「意味が近い言葉は周辺語も似ているはずです」の考え方で、つまりあるという単語の周辺にはどういった単語が出現しやすいか、という確率を考えることができます。

このような確率を、以下のような条件付き確率として softmax 関数を使ってモデル化します。

$$p(wo|wt) = \frac{exp(v'^{T}_{wo} \cdot v_{wt})}{\sum_{wv \in V} exp(v'^{T}_{wv} \cdot v_{wt})}$$

Exp$(x) = e^{x}$です。またv^{T}はvの転置、$v^{T} \cdot v$ はベクトルの内積です。wt は注目する単語、wo は wt の周辺の単語（添字の O は Output の 意味）V はボキャブラリ、v'_{wo}と
v_{wt}は単語を表すベクトルです。v は入力ベクトル v′は出力ベクトル。この式を計算すれば wt と wo の共起確率が計算できる。c 個の出力層の語群 wo1, wo2, …, woc が共起する確率は

$$p(wo1, wo2, \cdots, woc|wt) = \prod_{c=1}^{C} \frac{\exp(v'^{T}_{woc} \cdot v_{wt})}{\sum_{wv \in V} \exp(v'^{T}_{wv} \cdot v_{wt})}$$

この方法では計算量が非常に膨大になります。通常、ボキャブラリーの数が数万になるため、分母の計算に非常に時間がかかるのが理由です。なので、なんらかの工夫をしないととても実用には耐えられません。

3.1.2 高速化の手法「Negative Sampling」

word2vec では、より高速に学習させる手法として 2 つの手法がある。Negative Sampling と階層的ソフトマックスである。

Negative Sampling は wt の周辺範囲内にある単語については関連性の確率を高く、それ以外からランダムに選ばれた単語については確率を低くするように学習していく手法である。

着目する単語を wt、wo は wt の周辺の単語、wo が wt の wo は wt の周辺の単語$p(wo|wt)$と表すと、関連しない単語である確率は$1 - p(wo|wt)$となる。

$p(wo|wt)$の数式はシグモイド関数あるいはロジスティック関数である。0〜1 までの間でなだらかな曲線を描く関数で、確率の表現に使えるものである。

そして、関連する単語である確率

$$p(wo|wt) = \sigma(v_{wo}^{T} \cdot v_{wt}) = \frac{1}{1+\exp(-v_{wo}^{T} \cdot v_{wt})}$$

と関連しない単語である確率この２つの式から、以下のような関数を考える。

$$\mathrm{E} = -\log p(wo|wt) \prod_{wv \in V_{neg}} (1 - p(wo|wt))$$

V_{neg}は、関連性の無い単語の集合で、wt で指定された範囲の外にある単語から選ばれた単語です。

この式ですが、いわゆるロジスティック回帰による最尤推定法です。通常この手の問題は、尤度関数を最大にするようにパラメータの値を求めていくのですが、自然対数を取る数式になっているので、この関数の値を最小にする単語ベクトルの値を求めていく。

3.1.3　dmpv と DBoW

doc2vec は word2vec（Mikolov et al., 2013a）（Mikolov et al., 2013b）を応用した手法となっている。文書をベクトル化して、文書やテキストの分散表現を獲得する。そしてベクトル同士の類似度を測定して、文書分類や似た文書を探すことができる。具体的には word2vec の CBOW と skip-gram を対応して dmpv と DBoW という 2 つの手法が存在する。Word2Vec の CBoW における入力は、単語を one−hot 表現した単語 ID だけだったが、 Doc2Vec は、単語 ID にパラグラフ ID を付加した情報を入力とする。

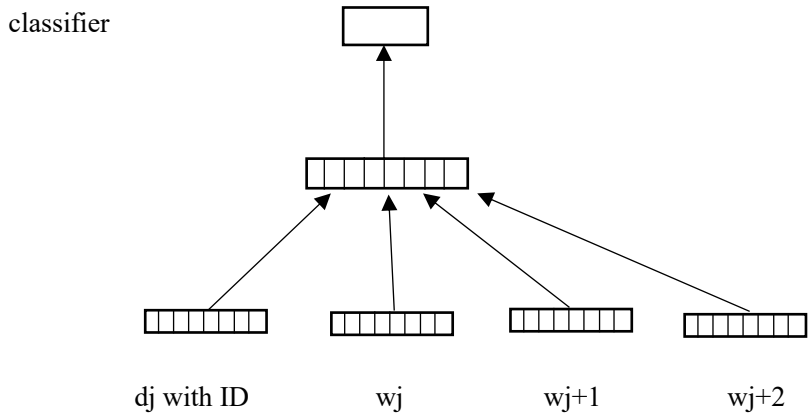

図 3.3　dmpv モデル

dmpv (Distributed Memory version of Paragraph Vector)は word2vec で使われる CBoW を応用したもの。中間層の入力は単語ベクトル $\boldsymbol{wj}$, $\boldsymbol{wj}$+**1** , … と 文書ベクトル $\boldsymbol{dj}$ 、入力ベクトルに単語列だけでなく、文書 ID を付与している。

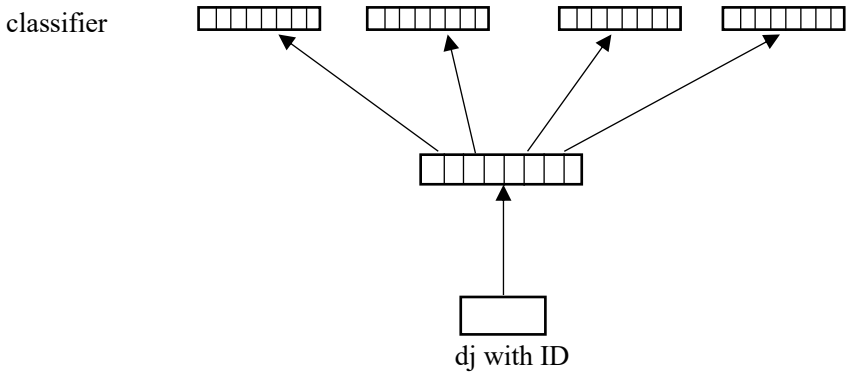

図 3.4　DBoW モデル

DBoW (Distributed Bag of Words version of Paragraph Vector)は word2vec で使われる Skip−gram を応用したものである。Skip−gram は CBoW とは逆に単語から周辺単語を予測するため、入力は単語であるが、DBoW では単語ではなく文書 ID を用いる。

DBoW は単語の順序を考慮しないシンプルなモデルで計算効率が良く、dmpv は DBoW と比べると少し複雑でより多くのパラメータが必要になる。

3.2　SVD

特異値分解(singular value decomposition)とは $M \times N$ の行列 X を以下のような 3 つの行列の積に分解する手法である。ここでは　$M < N$ を仮定する。

$$X = U\Sigma V^T$$

ここで X のランクを r とすると、行列 U は $M \times r$ であり、 X の列ベクトルが張る空間の正規直交基底となる。また行列 Σ は $r \times r$ の固有値を大きい順に並べた対角行列である。また行列 V^T は $r \times N$ の行列であり、 X の行ベクトルが張る空間の正規直交基底となっている。

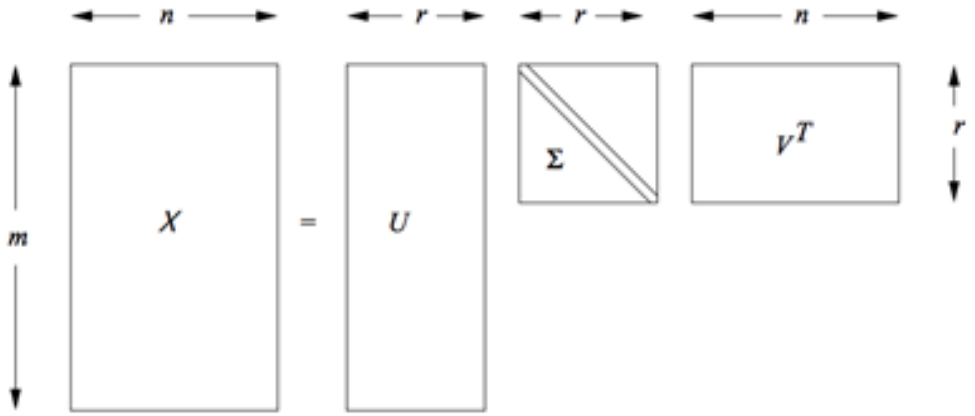

図 3.5　特異値分解

X をコーパスから作られる文書索引語行列（つまり行ベクトルが文書ベクトル）とし、行列 V の上位 k 列から作られる行列を V_k とすれば、XV_k によりコーパス内の文書が N 次元から k 次元へ縮約できる。

3.3　埋め込み表現の組み合わせ

文書 d の doc2vec による埋め込み表現を e、特異値分解による埋め込み表現（次元縮約ベクトル）を g としたとき、 e と g を連結したベクトル［ e ： g ］も文書 d の埋め込み表現と見なせる。

文書 d の bag-of-words のモデルによるベクトルを v としたとき、通常分類器の学習では特徴ベクトルとして v が用いられるが、本手法では v の代わりに、上記 3 つのベクトルを連結した［ v ： e ： g ］を特徴ベクトルとすることである（図 3.6 参照）。

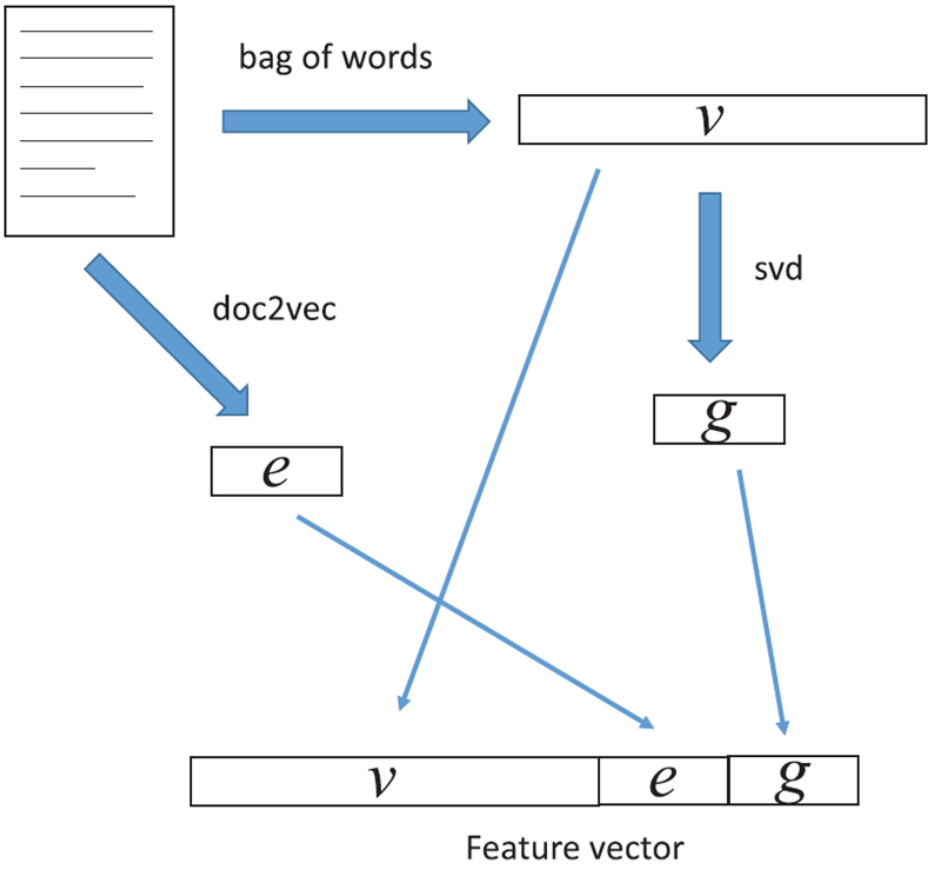

図 3.6　埋め込み表現の連結による特徴ベクトルの作成

第 4 章

実験

4.1　実験設定

実験データは以下で公開されている　Amazon Dataset の日本語文書を利用した。

https://www.uni-weimar.de/en/media/chairs/computer-science-department/webis/data/corpus-webis-cls-10/#webis-download

各領域に存在する文書数を表 4.1 に示す。

表 4.1：データ数

Domain	training	test	unlabeled
B (books)	2,000	2,000	169,780
D (dvd)	2,000	2,000	68,326
M (music)	2,000	2,000	55,892

実験ではある手法を利用して、文書を特徴ベクトルで表現する。次にその特徴ベクトルを用いて、ソース領域 S の training データから SVM の分類器を学習し、その分類器によりターゲット領域 T の test データを識別する。手法の評価は平均正解率で行う。

文書をどのように特徴ベクトルで表現するかが手法に対応する。BOW（bag-of-words）のモデルを用いて文書を特徴ベクトル v で表す手法をここでは「BOW」と呼ぶ。次に表 1 に示す全文書（305, 998 文書）から doc2vec により文書を 100 次元ベクトル e に埋め込み、文書の特徴ベクトルとして v と e の連結である[v ： e］用いた手法を「D2V」と呼ぶ。また次元縮約として特異値分解を用いて文書を 100 次元ベクトル¥(g ¥)に変換し、文書の特徴ベクトルとして¥(［v ： g］¥)を用いた手法を「SVD」と呼ぶ。提案手法は上記の v 、 e 、 g を連結したベクトル［v ： e ： g］を文書の特徴

ベクトルとしたものであり「D2V+SVD」と呼ぶ。

4.2　実験結果

実験結果を表 4.2 と図 4.1 に示す。

表 4.2：実験結果

DA	BOW	D2V	SVD	D2V+SVD
B → D	0.6980	0.7245	0.7220	**0.7360**
B → M	0.6935	0.7005	0.6885	**0.7050**
D → B	0.6660	0.6840	**0.7260**	0.7155
D → M	0.6910	0.6935	**0.7410**	0.7345
M → B	0.6235	0.6370	0.6835	**0.6845**
M → D	0.6855	0.6900	**0.7205**	0.7130
Average	0.6766	0.6883	0.7136	**0.7148**

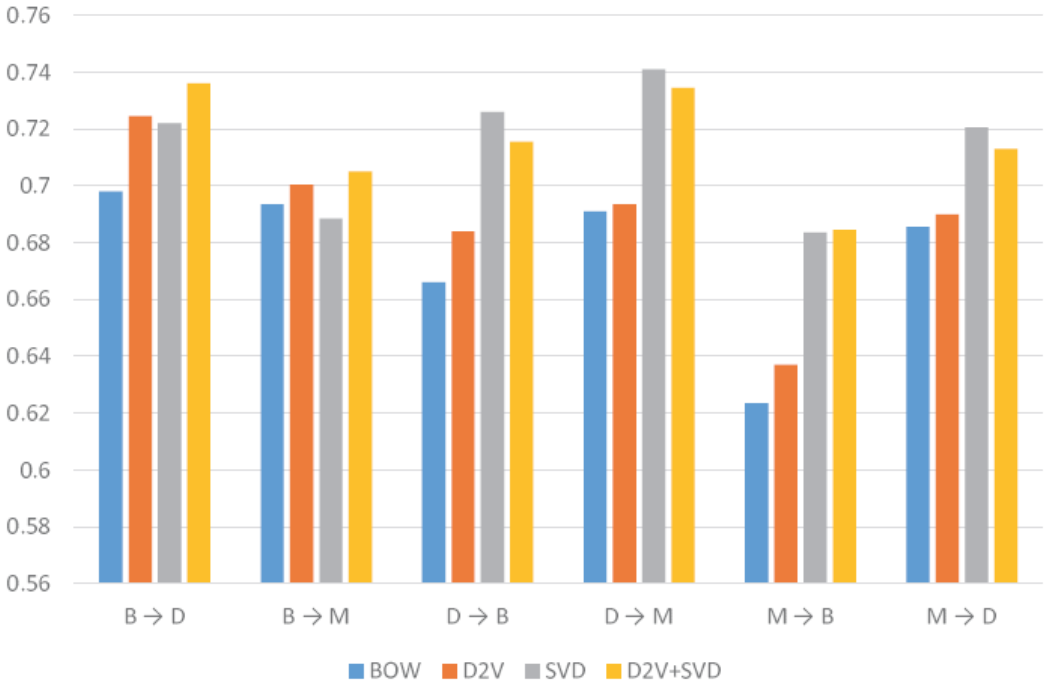

図 4.1：実験結果

埋め込み表現を利用した手法である D2V や SVD は、どちらもベースラインである BOW よりも高い精度を出している。また提案手法は更にそれらの手法よりも高い精度を出しており、提案手法の有効性が示されている。

第 5 章

考察

5.1　埋め込み表現の次元数

実験では doc2vec も特異値分解でも埋め込み表現の次元数は 100 に設定した。表 2 を見れば doc2vec の方は改善できる余地があることが予想できる。そこでここでは doc2vec による埋め込み表現の次元数を 50 次元、150 次元に変えて、上記実験を行った。結果を表 5.1 に示す。

表 5.1：doc2vec の次元数

DA	BOW	D2V (50 dim.)	D2V (100 dim.)	D2V (150 dim.)
B → D	0.6980	0.7165	**0.7245**	0.7090
B → M	0.6935	0.6950	0.7005	**0.7025**
D → B	0.6660	0.6680	**0.6840**	0.6740
D → M	0.6910	**0.7215**	0.6935	0.7090
M → B	0.6235	0.6300	**0.6370**	0.6335
M → D	0.6855	0.6885	**0.6900**	0.6790
Average	0.6766	0.6866	**0.6883**	0.6845

概ね次元数 100 のものが良い精度を出しているが、一部、50 次元や 150 次元のものが良い値を出している。最適な次元数はソース領域とターゲット領域のタイプによって異なる。最適な次元数の推定は今後の課題である。

5.2　埋め込み表現の構築元のコーパス

実験では埋め込み表現を作成する際に 3 つの領域全てのコーパスを利用している。表 1 を見ると、領域 books のコーパスが他と比べて明らかに大きい。そのため実験結果でもターゲット領域が books となる D → B や M → B の正解率の向上が大きい（表 5.2 参照）。

表 5.2：正解率の向上率

DA	BOW	D2V+SVD	向上率
B → D	0.6980	0.7360	1.054
B → M	0.6935	0.7050	1.017
D → B	0.6660	0.7155	**1.074**
D → M	0.6910	0.7345	1.063
M → B	0.6235	0.6845	**1.098**
M → D	0.6855	0.7130	1.040
Average	0.6766	0.7148	1.056

ここでは埋め込み表現を作成する際に利用するコーパスをソース領域とターゲット領域のものだけに設定した場合の実験を行い、全てのコーパスを利用する方がよいのか関連するコーパスだけを使った方が良いのかを考察する。ここでの実験では領域 dvd と領域 music の 2 つのコーパスから doc2vec や特異値分解による 100 次元の埋め込み表現を作成し、D → M や M → D に対して前章と同じ実験を行った。実験結果を表 5.3 に示す。

表 5.3：関連領域コーパスのみからの埋め込み表現の構築

DA	BOW	D2V using B,D,M	SVD using B,D,M	D2V+SVD using B,D,M	D2V using D,M	SVD using D,M	D2V+SVD using D,M
D → M	0.6910	0.6935	0.7410	0.7345	0.7085	0.7155	0.7170
M → D	0.6855	0.6855	0.6900	0.7130	0.6865	0.6990	0.6960
Average	0.6882	0.6895	0.7155	0.7238	0.6975	0.7073	0.7065

領域 dvd と領域 music の 2 つのコーパスから作った doc2vec による埋め込み表現は、全ての領域である領域 dvd と領域 music と 領域 books の 3 つのコーパスから作った doc2vec による埋め込み表現よりも高い精度を示した。一方、特異値分解による埋め込み表現は領域 books のコーパスを外したことで精度が下がった。つまり領域適応に

doc2vec を利用するときは関連するコーパスだけを使い、特異値分解を利用するときには全てのコーパスを利用するのが良いと言える。

最後に領域 dvd と領域 music の 2 つのコーパスから作った doc2vec による埋め込み表現と全ての領域のコーパスから特異値分解による埋め込み表現を組み合わせ、本手法を適用した結果を表 5.4 に示す。

表 5.4：関連領域コーパスのみからの埋め込み表現の構築(2)

DA	D2V+SVD (D2V using B,D,M SVD using B,D,M)	D2V+SVD (using D,M using B,D,M)
D → M	0.7345	0.7465
M → D	0.7130	0.7130
Average	0.7238	0.7298

領域適応 M → D は精度の変化はなかったが、領域適応 D → M において精度がさらに改善された。

第 6 章

結論

本論文では感情分析の領域適応のタスクに対して、文書に対する複数の複数特徴ベクトルを組み合わせて利用する手法を提案した。具体的には文書に対する bag of words モデルからのベクトル v に doc2vec による埋め込み表現 e と特異値分解による埋め込み表現 g を連結したベクトル［v ： e ： g］を特徴ベクトルとして用いる。実験では Amazon Dataset の領域 books, dvd, music の 3 つを利用して、計 6 通りの領域適応を行い、提案手法の有効性を示した。今後は埋め込む表現の最適な次元数や埋め込み表現を構築するコーパスの領域との関連を調査したい。

謝辞

本研究を進めるにあたり、多くのご指導、ご協力を頂いた指導教員の新納浩幸教授に感謝致します。また、日常の議論を通して多くの知識や示唆を頂いた佐々木稔講師、古宮嘉那子講師と新納研究室の皆様に感謝します。

参考文献

[1] John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP-2006*, pages 120-128.

[2] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölk opf,and Alex J Smola. 2006. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49-457.

[3] Daumé III, Hal, 2007. Frustratingly Easy Domain Adaptation. In *ACL-2007*, pages 256-263.

[4] Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupercised domain adaptation by backpropagation. In *ICML*, pages 1180-1189.

[5] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *ICML-11*, pages 513-520.

[6] Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical Insighs into document embedding generation. *arXiv preprint arXiv:1607.05368.*

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR Workshop paper*.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Adcances in neural information processing systems*, pages 3111-3119.

[9] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345-1359.

[10] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2015. Visual domain adaptain: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53-69.

[11] Anders Søgaard. 2013. *Semi-Supervised Learning and Domain Adaptation in Natural Language Processing*. Morgan & Claypool.

[12] Masashi Sugiyama and Motoaki Kawanabe. 2011. *Machine Learning in Non-Stationary Environments:* Introduction to Covariate Shift Adaptation. MIT Press.

[13] Baochen Sun and Kate Seanko. 2016. Deep coral: Correlation alignment for deep domain adaptation. *In Computer Vision-ECCV 2016 Workshops*, pages 443-450.

[14] Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of Frustratingly Easy Domain Adaptation. *AAAI*.

[15] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464.*