Paragraph Vectorへ埋め込む有効な付随情報の検討

橋戸　拓也† 　新妻　弘崇†† 　太田　学 †††

† 岡山大学工学部情報系学科　〒 700–8530 岡山県岡山市北区津島中 3–1–1
†† , ††† 岡山大学大学院自然科学研究科　〒 700–8530 岡山県岡山市北区津島中 3–1–1
E-mail: †pldu28hq@s.okayama-u.ac.jp, ††niitsuma@suri.cs.okayama-u.ac.jp, †††ohta@de.cs.okayama-u.ac.jp

あらまし　文章の特徴を表現する手法として Paragraph Vector が提案されている．Paragraph Vector は, 文章の特徴を表した数百次元程度の特徴ベクトルを獲得する手法で, この特徴ベクトルを用いると文章分類が高精度で実現できることが報告されている．一般的に, 文章は商品レビューの評価スコアや画像といった, 文章の内容を補助的に表す付随情報を伴うことが多い．我々は文章の付随情報に着目し, Paragraph Vector を拡張した ScoreSent2Vec を提案した．ScoreSent2Vec は付随情報の影響を受けた文章の特徴ベクトルを生成する．ScoreSent2Vec で生成した特徴ベクトルを用いると, Paragraph Vector で生成した特徴ベクトルよりも高精度な分類が実現できることを示した．本稿では, ScoreSent2Vec で用いる有効な付随情報の選択基準を示すために, 付随情報のエントロピーに着目する．実験ではエントロピーに差がある付随情報を複数用意し, 分類精度とエントロピーを比較することで有効な付随情報について検討する．また, ScoreSent2Vec に存在する複数のパラメータを調整することで, 各パラメータが分類精度に与える影響について考察する．

キーワード　word2vec, ニューラルネットワーク

1. はじめに

近年, Mikolov らによって提案されたニューラルネットワークである word2vec [1] [2] が注目されている．word2vec を用いると, 単語の特徴を比較的低次元な数値ベクトルで表現することができる．また, 意味の近い単語から生成されたベクトルは類似したベクトルになる．word2vec は文章中のある単語を単語の前後関係から予測するという問題の解を, ニューラルネットワークに学習させることで, そのニューラルネットワークの中間層の値を単語の特徴ベクトルとして抽出するものである．文章中の単語を予測する方法としては, 中心の単語から周辺の単語を予測するものと, 周辺の単語から中心の単語を予測するものがある．それぞれ Skip-gram モデルと Continuous Bag-of-Words (CBOW) モデル [1] [2] と呼ばれている．

従来, 文章の特徴を表現する方法として, TF-IDF や Bag-of-Words (BOW) のように単語の出現頻度を用いたものがある．しかしながら、これらには語順を考慮できないという問題がある．他にも, 同義語に対して異なるベクトルを割り当てるといった問題もある．word2vec はこれらの問題の解決に有効であるものの, 文章の特徴ベクトルを生成することはできない．そこで, word2vec と同じ原理で文章の特徴ベクトルを生成できるよう拡張したのが Paragraph Vector [3] である．映画のレビュー文章から, レビュアがつけた映画の評価値を推定する問題 [4] において, Paragraph Vector は従来の BOW などを使った手法よりも高精度な推定を実現できたことが報告されている [3].

我々は [5] で文章に付随する情報に着目し, Paragraph Vector を拡張した ScoreSent2Vec を提案した．ここで付随する情報とは, 文章の内容を補助的に表現した情報のことである．例えば, Amazon.com(注1) や価格.com(注2) などの商品のレビュー文章には, その商品の評価スコアがレビュー文章と共に記載されている．評価スコアとは, 通常評価を星の数で表したもので 1 から 5 までの 5 段階で表される．この場合, 評価スコアはレビュー文章の付随情報とみなせる．ScoreSent2Vec は, Paragraph Vector に付随情報を予測するニューラルネットワークを追加するという拡張によって, 付随情報の影響を受けた文章の特徴ベクトルを生成する．我々は ScoreSent2Vec により, 従来の Paragraph Vector よりも高精度な分類が可能であることを 2 つの実験で示した [5]．1 つ目は, Brown Corpus [6] に含まれる英文をカテゴリ毎に分類する実験である．カテゴリは news, religion, hobbies, science fiction, romance, humor の 6 種類である．付随情報を助動詞 can, could, may, might, must, will の各出現回数を並べた 6 次元ベクトルとして正解率を算出したところ, Paragraph Vector が 0.487 であったのに対し, ScoreSent2Vec は 0.494 となった．2 つ目は, 画像付ツイートの分類実験である．ツイートが柴犬の正解画像かどうかをツイート文を用いて分類した．付随情報を画像の色ヒストグラムとして F-measure を算出したところ, 色ヒストグラムの bin が 4 の時, Paragraph Vector が 0.584 であったのに対し, ScoreSent2Vec は 0.609 であった．

我々は [5] で提案した ScoreSent2Vec により, 付随情報が分類精度を向上させることは示した．しかし, どのような付随情報が分類精度に影響を与えるか等有効な付随情報については未知である．そこで本稿では付随情報のエントロピーと分類精度の関係を示すことで, ScoreSent2Vec に有効な付随情報の選択

(注1)：http://www.amazon.com
(注2)：http://kakaku.com

基準を示す．また, Paragraph Vector, ScoreSent2Vec には調整可能なパラメータが存在するため, パラメータを調整して分類精度を比較することでパラメータが分類精度に与える影響についても考察する．

2. 関連研究

ScoreSent2Vec の基礎である word2vec と Paragraph Vector について説明する．word2vec は単語を特徴ベクトルとして表し, Paragraph Vector は文章を特徴ベクトルとして表す．

Mikolov らは word2vec で生成した 2 つの単語ベクトルの差は, 2 つの単語の関係を表現していると主張した．以下の式を例に単語ベクトルについて説明する．

$$\text{king} - \text{man} + \text{woman} \tag{1}$$

式 (1) は king, man, woman の単語ベクトルを用いた演算で, この式の解は man と king の関係を woman に適用したものと言える．Mikolov らは, word2vec で生成した単語ベクトルを用いてこの演算をした場合, 答えは queen の単語ベクトルになると主張した．そこで, Athens − Greece + Norway の解が Oslo になれば正解といったテストを行った結果, 他のニューラルネットワークを使用したモデルより word2vec を用いて分類した方が高い正解率を実現した [2].

word2vec は, 特徴ベクトルの生成にコーパスが必要である．このコーパス中の単語の前後関係を用いてニューラルネットワークの学習を行う．ここで, Mikolov らは word2vec に用いるコーパスの単語数が多い場合, 単語のベクトルの次元数も大きくすることで, 式 (1) のような演算の精度が上がることを示した [2]．このように, より高精度な分類を行うためには単語ベクトルの次元数を調整する必要がある．これは Paragraph Vector, ScoreSent2Vec でも同じことが言える．これらの特徴ベクトル生成手法には, 単語ベクトルの次元数以外にも出現回数が低頻度の単語を考慮しないための頻度の閾値などの調整可能なパラメータが存在する．そこで本稿では, Paragraph Vector や ScoreSent2Vec のパラメータを調整して分類精度を比較する．

2.1 word2vec

Mikolov らは word2vec を実現するニューラルネットワークの構造として Skip-gram モデルと CBOW モデルを提案している [1] [2]．以下ではこれら 2 つのモデルについて説明する．

Skip-gram モデル: 図 1 に Skip-gram モデルのニューラルネットワークを示す．このモデルは入力層, 中間層, 出力層からなり, 文章中のある単語 $w(t)$ を入力とし, その前後の単語 $w(t-c), ..., w(t-1), w(t+1), ..., w(t+c)$ を出力とするニューラルネットワークである．図 1 から図 6 は $c = 2$ としたときのニューラルネットワークである．c は同じ文脈として考慮する前後の単語数を示す．

CBOW モデル: 図 2 に CBOW モデルのニューラルネットワークを示す．このモデルは Skip-gram モデルと同様に入力層, 中間層, 出力層からなるが, 入力と出力が Skip-gram モデルとは逆となる．出力は中心の単語 $w(t)$ であり, 入力はその前後の

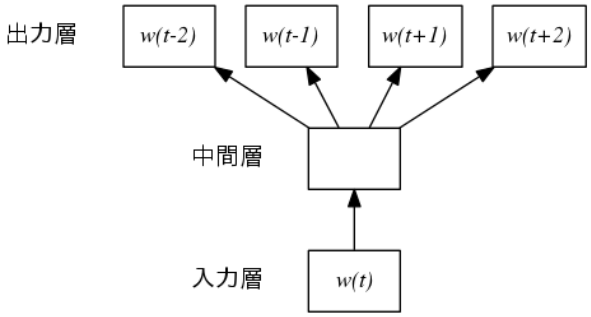

図 1 Skip-gram モデル

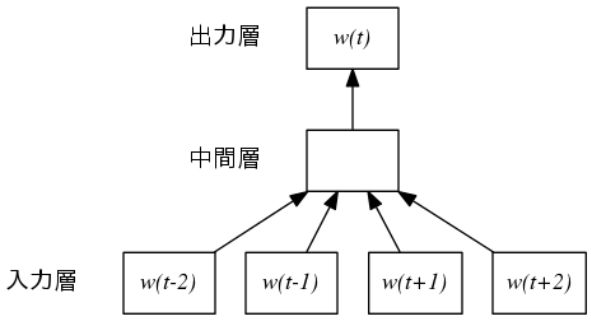

図 2 CBOW モデル

単語 $w(t-c), ..., w(t-1), w(t+1), ..., w(t+c)$ とするニューラルネットワークである．つまり, Skip-gram モデルとは反対に, 周辺の単語から中心にある単語を推定する問題をニューラルネットワークに学習させる．

黒崎ら [7] は文章中に含まれる顔文字の感情分類を行うために word2vec を利用した．Twitter に投稿された文章を対象とし, 顔文字と感情毎に分類された語との類似度を計測した．ここで, 感情は喜び, 悲しみ, 落胆, 安心, 驚き, 怒りの 6 種類に分類されている．顔文字と最も高い類似度を示す語が属する感情をその顔文字が表す感情とした．顔文字を紹介しているウェブサイト上の分類を正解とし, 分類結果の F-measure を算出したところ, 最も高かったのは“ 喜び ”の 0.847 で, 6 種類の感情全体では 0.746 であった．

2.2 Paragraph Vector

Paragraph Vector は word2vec の拡張で, 単語ではなく文章の特徴ベクトルを生成する手法である．Paragraph Vector にも word2vec と同様に 2 つのモデルがある．Skip-gram モデルを拡張した Paragraph Vector with Distributed Bag of Words（PV-DBOW）モデルと CBOW モデルを拡張した Paragraph Vector with Distributed Memory（PV-DM）モデルである．以下ではこれら 2 つのモデルについて説明する．

PV-DBOW モデル: 図 3 に PV-DBOW モデルのニューラルネットワークを示す．このモデルは word2vec の Skip-gram モデルを拡張したものである．最初に word2vec の Skip-gram モデルを使って学習したニューラルネットワークを作成する．次に Skip-gram モデルのニューラルネットワークの入力層を, 文

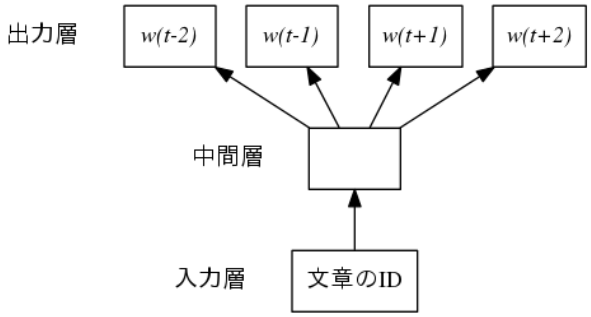

図 3　PV-DBOW モデル

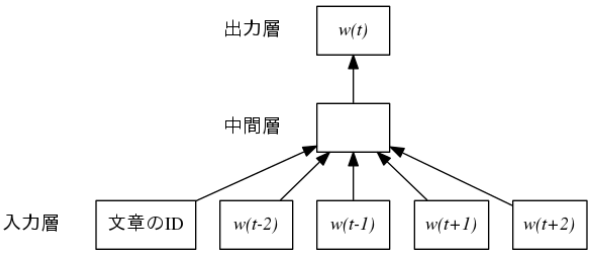

図 4　PV-DM モデル

章の ID を入力するネットワークと入れ替える. 文章の ID のみを, word2vec と同様の方法で目的の文章に対して学習することで, 文章の特徴ベクトルを得ることができる.

PV-DM モデル:　図 4 に PV-DM モデルのニューラルネットワークを示す. このモデルは, word2vec の CBOW モデルを拡張したモデルである. PV-DBOW モデルと同様に, 初めに word2vec の CBOW モデルを使って学習したニューラルネットワークを作成する. 次に, 文章の ID を入力するネットワークを入力層に追加する. この追加されたネットワーク部分のみを word2vec と同様の方法で目的の文章に対して学習することで, 文章の特徴ベクトルを得ることができる.

Paragrh Vector を用いた研究も多数報告されている. 中野ら [8] は, 提題表現に基づいた重要段落の抽出に Paragraph Vector を用いた. 提題表現とは, 文の主題を取り上げる表現である「〜は」の形式を典型とする. 中野らの提案手法では, 文章中に記述してある語をベクトル化し, 段落ごとに得たベクトルとの内積を計算することで重要段落を得る. 毎日新聞コーパス (1998-99 年) 及び日経新聞コーパス (1998 年) からニュース報道記事に該当するものを実験データとしたところ, 毎日新聞記事では 61.2 %, 日経新聞記事では 77.9 %の抽出精度を得た.

佐藤ら [9] はウェブ上の有害な文書を分類するために Paragraph Vector を用いた. 実験では, 有害文書と無害文書をそれぞれ 10 万件ずつ用いて評価実験を行った. 佐藤らの提案手法の中では, PV-DM モデルを拡張した PV-CBOW(Continuous Bag-of-Words of Paragraph Vectors) モデルの F-measuer が最も高く, ベクトルの次元数が 200 の時に 0.9431 であった.

2.3　Paragraph Vector と word2vec の関係

Paragraph Vector は word2vec の文章への拡張である. しかし, 実装の上では Paragraph Vector の中で word2vec を用いている. 図 2, 図 4 を用いて説明する. 図 2 では, ある単語 $w(t)$ を周辺の単語 $w(t-2), w(t-1), w(t+1), w(t+2)$ から予測することで, 中間層の値を $w(t)$ のベクトルとした. つまり, $w(t-2), w(t-1), w(t+1), w(t+2)$ を入力としたニューラルネットワークにおいて, $w(t)$ が出力となるように重みを調整していくのが図 2 である. こうして全ての単語に対して重みが調整され, 各単語のベクトルが生成される. ここで, 図 4 の $w(t-2), w(t-1), w(t+1), w(t+2)$ は, word2vec で生成された単語のベクトルを用いる. 既に重みが調整された単語のベクトルに対して, 文章の情報を追加して重みを調整するのが Paragraph Vector である.

2.4　ScoreSent2Vec

Paragraph Vector の拡張である ScoreSent2Vec [5] のモデルについて説明する. ScoreSent2Vec とは, Paragraph Vector に付随情報を予測するニューラルネットワークを追加する拡張をしたものである. ここで, 付随情報とは比較的低次元の数値ベクトルで表現できるものを対象とし, そのベクトルを本稿では score vector と呼ぶ. ScoreSent2Vec も Paragraph Vector と同様に 2 つのモデルが存在する. PV-DBOW モデルを拡張した Scored Paragraph Vector with Distributed Bag-of-Words（SPV-DBOW）モデルと PV-DM モデルを拡張した Scored Paragraph Vector with Distributed Memory（SPV-DM）モデルである. 以下ではこれら 2 つのモデルについて説明する.

SPV-DBOW モデル:　図 5 に SPV-DBOW モデルのニューラルネットワークを示す. このモデルは Paragraph Vector の PV-DBOW モデルを拡張し, score vector のベクトル値を予測するニューラルネットワークを追加したモデルである. PV-DBOW モデルと同様に文章の ID から, ある単語と同じ文脈内に含まれる周辺の単語と score vector の値を同時に予測する問題をニューラルネットワークに学習させることで, 付随情報を考慮した文章のベクトル表現を得ることができる.

SPV-DM モデル:　図 6 に SPV-DM モデルのニューラルネットワークを示す. このモデルは, Paragraph Vector の PV-DM モデルを拡張し, score vector のベクトル値を予測するニューラルネットワークを追加したモデルである. PV-DM モデルを基礎としており, それ以外は SPV-DBOW モデルと同様の処理を行う.

なお, ScoreSent2Vec のソースコードは次のサイトで公開している.

`https://github.com/niitsuma/ScoreSent2Vec`

3.　有効な付随情報

本稿では, 文章の付随情報のエントロピーと分類精度を比較することで, ScoreSent2Vec に用いる有効な付随情報の選択基準を示す. すなわち score vector の選択基準を示すということである. エントロピーとは情報の曖昧さを表す尺度で, 平均情報量とも呼ばれる. ある事柄の発生確率が全て同じときに最大となり, 発生確率に偏りが大きいほどエントロピーは小さくな

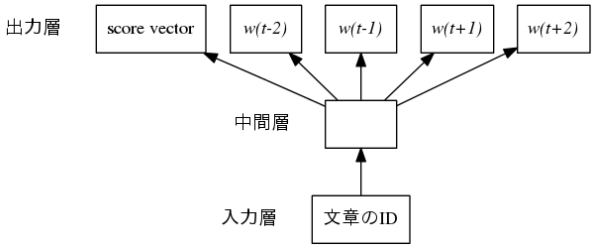

図 5　SPV-DBOW モデル

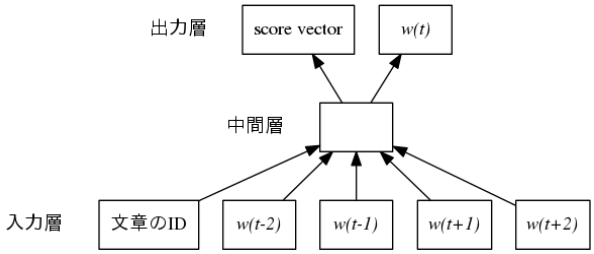

図 6　SPV-DM モデル

る. score vecor のエントロピー H は以下の式で表される.

$$H = -\sum_{l \in L, v \in V} P(l,v) * \log_2 P(l,v) \qquad (2)$$

ここで, L は分類したいラベルの集合, V はとりうる score vector の全集合である. $P(l,v)$ は分類対象の全ての文数で, score vector が v で, かつラベルが l である文の数割ることで算出する.

4. 節では, 複数の付随情報を用いて実験を行う. 式 (2) で各付随情報のエントロピーを算出し分類精度と比較することで, 分類に有効な付随情報の選択基準を示す.

また, 有効なモデル選択基準として幅広い分野で利用されている赤池情報量基準 (AIC) , ベイズ情報量基準 (BIC) も算出し, 分類精度と比較する.

4. 評価実験

2 種類の実験により有効な付随情報と有効なパラメータについて考察する. 1 つは, 映画のレビュー文から作られたコーパス, Stanford Sentiment Treebank Dataset [10] (SSTD) の文を Positive や Negative といった評価ラベル毎に分類する実験である. もう 1 つは, Brown Corpus 中の英文を news や religion といったカテゴリに分類する実験である. 本実験で調整するパラメータは以下の 3 種類である.

- d-size
  生成するベクトルの次元数を示す.
- min-count
  単語の出現回数でこの値より少ないものを除去する.
- scale
  score vector のスケーリングを行う. score vector と scale の積を用いる.

表 1　評価ラベル毎の文数

評価ラベル	トレーニングデータ	テストデータ
Very Negative	1,092	279
Negative	2,218	633
Neutral	1,624	389
Positive	2,322	510
Very Positive	1,288	399
all	8,544	2,210

なお, scale は ScoreSent2Vec にのみ用いる. word2vec は, トピックモデリングに特化した python のライブラリである gensim(注3) を用いて実装した. Paragraph Vector の実装は klb3713 の sentence2vec(注4) を用いた. ScoreSent2Vec は [5] で実装したものを使用した.

4.1 SSTD のレビュー分類実験

SSTD に含まれるレビュー文を評価ラベルごとに分類する実験を行う. 以下この実験をレビュー分類実験とする. SSTD は, Amazon Mechanical Turk(注5) を利用して, 文章に評価ラベルを付与した構文木コーパスである. Amazon Mechanical Turk とは, ソフトウェアに実行させるより人間が行うほうが効率的であると思われる作業を, 開発者が Web 上に掲示することで代行を依頼できるサービスである. SSTD は映画のレビューより抽出された単文から構成され, レビュー各文に対して Very Negative, Negative, Neutral, Positive, Very Positive の 5 種類の評価ラベルが付与されている. レビュー文は予めトレーニングデータとテストデータに分けられている. それぞれのレビュー文数は, トレーニングデータが 8,544 件, テストデータが 2,210 件であり, レビュー文は重複しない. 表 1 にトレーニングデータとテストデータのラベル毎の文数を示す.

初めにレビュー各文を以下の 3 種類の特徴ベクトルで表現する.

- Paragraph Vector で特徴ベクトル化
- Paragraph Vector で生成したベクトルに score vector を連結した特徴ベクトル
- ScoreSent2Vec で特徴ベクトル化

これらの特徴ベクトルを用いて分類を行った結果を, ここではそれぞれ Paragraph Vector, Concatenate, ScoreSent2Vec と呼ぶ. Concatenate とは Paragraph Vector と score vector を連結したもので, 例えば Paragraph Vector の d-size が 100 の場合, score vector の 6 次元ベクトルと組み合わせて 106 次元の特徴ベクトルとする. 本実験では, ScoreSent2Vec の付随情報は, 特定の 6 語の出現回数とする. 分類精度と付随情報のエントロピーを比較するために, エントロピーに差がある 6 語の組み合わせを以下の 3 通り用意した. 1 つは助動詞 can, could, may, might, must, will, 1 つは映画と関連の強い単語 actor, cast, film, scene, sound, story, 1 つは評価などに使われる語

(注3) : https://radimrehurek.com/gensim
(注4) : http://github.com/klb3713/sentence2vec
(注5) : https://www.mturk.com/mturk/welcome

表 2　レビュー分類実験の付随情報とエントロピー, AIC, BIC

単語群名	単語	エントロピー	AIC	BIC
助動詞群	can, could, may, might, must, will	3.04	5456.72	5467.42
特徴語群	actor, cast, film, scene, sound, story	3.27	5429.10	5444.84
評価語群	but, good, great, nice, no, not	3.60	5424.51	5454.98

表 3　パラメーター一覧

パラメータ	値
d-size	100, 200, 400
min-count	5, 10, 20
scale	0.01, 0.1, 1.0

but, good, great, nice, no, not である．これらの単語の組み合わせをそれぞれ助動詞群, 特徴語群, 評価語群と呼ぶ. 実験では, 単語群毎に各単語の出現回数を並べた 6 次元ベクトルを score vector とする．よって助動詞群の場合の score vector は以下のようになる.

$$\begin{pmatrix} (\text{can の出現回数}), (\text{could の出現回数}), \\ (\text{may の出現回数}), (\text{might の出現回数}), \\ (\text{must の出現回数}), (\text{will の出現回数}) \end{pmatrix} \quad (3)$$

表 2 に 3 通りの単語群とそのエントロピー, AIC, BIC を示す．エントロピーは式 (2) を用いて算出した．AIC, BIC は scikit-learn [11] のライブラリ関数である LassoLarsIC を使用して算出した．また, AIC と BIC を計算するために LassoLarsIC を使用した関係上, 分類する 5 種類の評価ラベル Very Negative, Negative, Neutral, Positive, Very Positive をそれぞれ 0, 1, 2, 3, 4 の値で表現した.

次にトレーニングデータを用いて生成した特徴ベクトルを用い, 各特徴ベクトル毎に Logistic Regression で分類器の学習を行う．Logistic Regression は scikit-learn のライブラリ関数を使用する．最後にテストデータを用い生成した特徴ベクトルを学習済みの分類器で分類し正解率を算出する．正解率とは, テストデータの全レビュー文数で正しく評価ラベルが付与されたレビュー文数を割ることで算出する.

表 3 に特徴ベクトル生成時に調整したパラメータの値を示す．パラメータの調整は, d-size, min-count, scale の 3 つのパラメータを各 3 種類ずつの計 9 通り行う．表 4 に助動詞群, 特徴語群, 評価語群それぞれで生成した特徴ベクトルを用いて分類した時の正解率とパラメータを示す．ここで示す正解率は Paragrph Vector, ScoreSent2Vec のパラメータ調整を行い, 正解率が最も高かったものである.

4.2　Brown Corpus のカテゴリ分類実験

ここでは Brown Corpus に含まれる英文がどのカテゴリに属するかを分類する実験を行う．以下この実験をカテゴリ分類実験とする．Brown Corpus は, 言語研究のための英文のコーパスで, 1961 年にブラウン大学で作成された．Brown Corpus の英文は全て news, religion などのカテゴリに分類されている．このカテゴリの内, news, religion, hobbies, science fiction, romance, humor の計 6 カテゴリに含まれる 16,964 文を実験対象とする．表 5 に各カテゴリ毎の文数を示す.

表 4　レビュー分類実験の付随情報別正解率

付随情報	特徴ベクトル生成手法	パラメータ			正解率
		d-size	min-count	scale	
-	Paragraph Vector	400	5	-	0.304
助動詞群	Concatenate	400	5	0.1	0.310
	ScoreSent2Vec	400	5	0.1	0.316
特徴語群	Concatenate	400	5	0.1	0.307
	ScoreSent2Vec	400	5	0.01	0.313
評価語群	Concatenate	100	10	0.1	0.305
	ScoreSent2Vec	400	5	1.0	0.315

表 5　カテゴリ毎の文数

カテゴリ	文数
news	4,623
religion	1,716
hobbies	4,193
science fiction	948
romance	4,431
humor	1,053
all	16,964

表 6　カテゴリ分類実験の付随情報とエントロピー, AIC, BIC

単語群名	単語	エントロピー	AIC	BIC
助動詞群	can, could, may, might, must, will	3.26	17348.92	17395.35
5W1H 群	what, when, where, who, why, how	3.12	17586.84	17625.53
頻出単語群	back, even, first, last, long, new	3.07	17477.49	17523.99

初めに 4.1 節と同様に, 各英文を 3 種類の特徴ベクトルとして表現する．それぞれの特徴ベクトルを用いて分類を行った結果を Paragraph Vector, Concatenate, ScoreSent2Vec とする．ここで本実験の score vector について説明する．本実験でも 4.1 節と同様に ScoreSent2Vec の付随情報を 6 語の出現回数とし, 6 語の組み合わせを 3 通り用意した．1 つは助動詞, can, could, may, might, must, will, 1 つは 5W1H, what, when, where, who, why, how, 1 つは 出現回数が多かった語, back, even, first, last, long, new である．これらの単語の組み合わせをそれぞれ助動詞群, 5W1H 群, 頻出単語群と呼ぶ. 表 6 に 3 通りの単語群とそのエントロピー, AIC, BIC を示す．エントロピーは式 (2) を用いて算出した．また, AIC と BIC は 4.1 節と同様に LassoLarsIC を使用して算出した．LassoLarsIC を使用するために humor, hobbies, news, romance, religion, science fiction をそれぞれ 0, 1, 2, 3, 4, 5 の値に相当するとした．このそれぞれのカテゴリへの 0, 1, 2, 3, 4, 5 の値の割り当て方は，AIC, BIC が最小となるような割り当てを選択した.

生成した特徴ベクトルを用いて分類を行う．5 分割交差検定で Logistic Regression を用いて分類し, 正解率を算出する．ここで正解率とは, 実験対象の全英文数 16,964 で正しいカテゴリ

表 7　パラメーター一覧

パラメータ	固定値	変動値
d-size	100	50, 80, 100, 200, 300, 400
min-count	5	5, 10, 20, 35
scale	0.1	0.01, 0.1, 1, 10

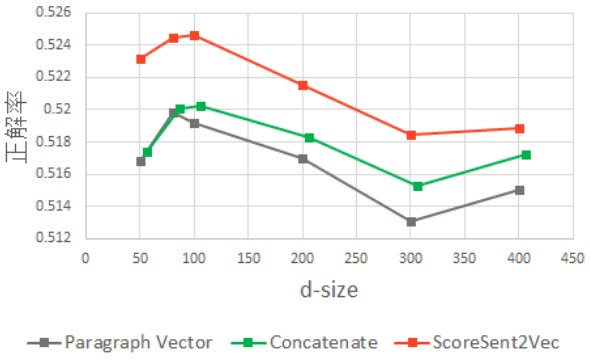

図 7　特徴ベクトル生成手法の正解率の比較

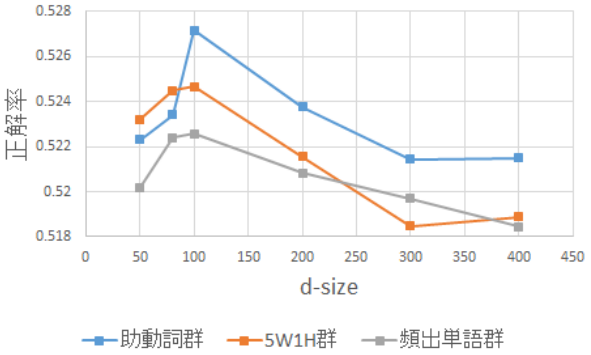

図 8　d-size と正解率の関係

に分類された英文数を割ることで算出する．

本実験では，d-size, min-count, scale の 3 つのパラメータの内，2 つを固定し残り 1 つを変化させたときの正解率を比較する．表 7 にパラメータを固定するときの値と変化させるときの値を示す．図 7 に付随情報を 5W1H 群として d-size のみを変化させたときの各特徴ベクトルの正解率を示す．この結果より，ScoreSent2Vec の付随情報の種類とパラメータを変えて正解率を比較する．

図 8, 図 9, 図 10 に，d-size のみ，min-count のみ，scale のみをそれぞれ変化させたときの付随情報の種類毎の正解率を示す．ここで，図 10 の横軸は対数スケールで表示している．

5.　考　　察

ここで，付随情報とエントロピーの関係から有効な付随情報について考察する．また，ScoreSent2Vec のパラメータが分類精度に与える影響について考察する．

5.1　有効な付随情報

3 種類の付随情報のエントロピー，AIC, BIC を比較し，分類精度との関係から有効な付随情報について検討する．4.1 節ではレビュー分類実験の付随情報のエントロピー，AIC, BIC と正解

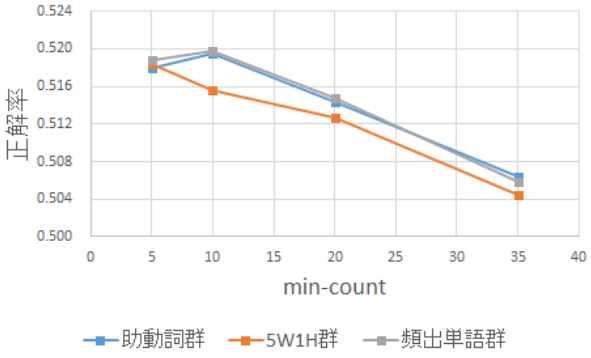

図 9　min-count と正解率の関係

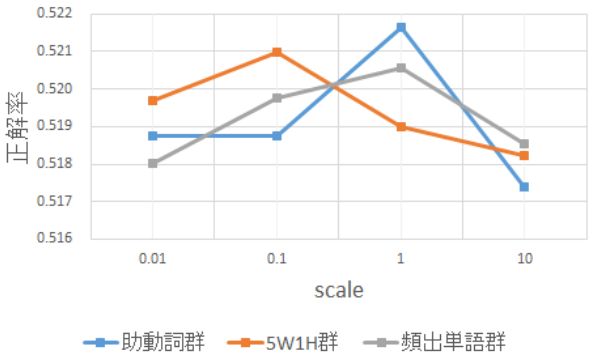

図 10　scale と正解率の関係

率の関係を示した．表 2, 表 4 から，何れの付随情報を用いた場合も，ScoreSent2Vec の方が Paragraph Vector, Concatenate よりも高い正解率となったことが分かる．次に，エントロピー，AIC, BIC と正解率を比較する．正解率は高いものから助動詞群，評価語群，特徴語群となる．エントロピーは高いものから評価語群，特徴語群，助動詞群，AIC は助動詞群，評価語群，特徴語群，BIC は助動詞群，特徴語群，評価語群となり，AIC とのみ正解率との大小関係が一致した．

4.2 節では，カテゴリ分類実験の付随情報のエントロピーと正解率の関係を示した．図 8, 図 9, 図 10 を比較すると，何れの付随情報でも d-size を調整したときに最も高い正解率を示した．3 種類の付随情報の中で最も高い正解率を示したのは助動詞群で，5W1H 群，頻出単語群と続く．表 6 より，エントロピーは高いものから助動詞群，5W1H 群，頻出単語群，AIC は 5W1H 群，頻出単語群，助動詞群，BIC は 5W1H 群，頻出単語群，助動詞群となっている．エントロピーのみ正解率との大小関係が一致した．また，AIC, BIC は一般的に値が低いほど良いモデルであるとされている．助動詞群は最も正解率が高く，かつ AIC, BIC は共に最小となることからこの性質に一致する．また，どの実験においても概ね助動詞群が良い精度を与えている．

ここで，2 つの実験の差分について述べる．1 つはデータ数である．カテゴリ分類実験で使用する英文が 16,964 文なのに対し，レビュー分類実験で使用する英文は 10,754 文とカテゴリ分類実験の 3 分の 2 以下の文数である．このため，レビュー分類実験で生成した特徴ベクトルは，分類に有効な差を持たなかっ

た可能性がある．もう 1 つは, 分類ラベルの性質の違いである．カテゴリ分類実験は news や religion といったカテゴリに分けられており, それぞれが各カテゴリの差分や類似度は明確ではない．しかし, レビュー分類実験の評価ラベルは Very Negative と Negative, Very Positive と Positive は関係が深く, Neutral は他のラベルと類似しないといったように, ラベル間の関係がカテゴリ分類実験と異なる．これらの違いから, エントロピーや AIC, BIC と正解率の関係に差が出たと考える．また, どちらの実験でも助動詞群を付随情報とした場合の精度が高かったことから, 助動詞を他と区別するような何らかの基準を検討する余地がある．そのような基準を使えば助動詞よりも良い付随情報が得られる可能性がある．

5.2 パラメータと分類精度

ScoreSent2Vec のパラメータ d-size, min-count, scale を調整し, 分類精度を比較した．4.2 節では特定のパラメータのみを調整して正解率を算出した．図 8 より, d-size についての考察を述べる．d-size を 50 以上の範囲について調べた結果は, 助動詞群が概ね高い正解率となった．また, 全ての付随情報に共通した点として, d-size は 100 を境に大きい程概ね正解率が下がることが挙げられる．これは, コーパスに対して適切な次元数を設定しないと分類精度に影響が出るという Mikolov らの指摘と合致する．図 9 より, min-count の考察を述べる．全ての付随情報に共通して言えることは, min-count は 10 を越えると正解率が下がるということである．min-count を高くすると除去される単語が多くなり, その結果単語や文の特徴を示す重要な単語も除去されたため正解率に影響が出たと考える．図 10 より, scale の考察を述べる．全ての付随情報に共通した点として, scale が大きすぎると正解率が下がることが言える．また, scale は小さすぎても正解率が下がる傾向にある．具体的には score vector の各要素の平均値が 0.01 から 10 程度となるような scale が最も良い精度を与えると言える．ScoreSent2Vec は, score vector をニューラルネットワークに埋め込むことで付随情報を反映させた文章の特徴ベクトルを生成する．つまり, 他の単語や文章のベクトルと大きく異なる値を取ると影響が強く出る．単語や文章のベクトルの各要素は絶対値が 1 以下の数値であるため, scale は 1 以下が良いと考えられる．

6. ま と め

本稿では ScoreSent2Vec へ埋め込む有効な付随情報及び有効なパラメータについて検討した．検討材料として SSTD のレビュー文を評価ラベル毎に分類する実験, Brown Corpus 中の英文をカテゴリ毎に分類する実験を行った．

2 種類の実験により, 分類精度は付随情報によって異なり, その差が付随情報によるエントロピーに関わる可能性を示した．カテゴリ分類実験により, 付随情報のエントロピーが高いほど, また, AIC, BIC が低いほど分類精度が向上することが分かった．しかし, レビュー分類実験ではその傾向を確認できなかった．

ScoreSent2Vec に有効なパラメータについては, 約 17,000 文のコーパスに対する本稿で示したようなカテゴリ分類実験の場合, d-size が 100 前後, min-count が 10 前後, scale が 0.01 から 10 の間にすることで高精度な分類が可能であることを示した．

今後の課題として, 様々なデータセットを用いた分類実験を実施する必要がある．本研究で考察したエントロピー, AIC, BIC と分類精度の関係をデータセットを変えて確認することで, ScoreSent2Vec に有効な付随情報の選択基準を明確にできると考える．また, 本研究で行ったパラメータ調整は類似研究と比較して十分とは言い難い．そのため, より効率的なパラメータ調整も課題である．

文 献

[1] Mikolov, T., Sutskever, I., Chen, K., Corrado G. and Dean, J.: Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems*, pp. 3111–3119 (2013).

[2] Mikolov, T., Chen, K., Corrado G. and Dean, J.: Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*, pp. 1–12 (2013).

[3] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *CoRR*, abs/1405.4053, pp. 1–9 (2014).

[4] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manningand, C. D., Ng, A. Y. and Potts, C.: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, Association for Computational Linguistics, pp. 1631–1642 (2013).

[5] 橋戸拓也, 新妻弘崇, 太田学: Paragrph Vector への追加情報の効率的な埋め込み, 情報処理学会研究報告, Vol. 2015-DBS-162, No. 11, pp. 1-8, 2015.

[6] Francis, W. N., Kucera, H.: *Brown Corpus Manual*, Brown University (1979).

[7] 黒崎優太, 高木友博: Word2Vec を用いた顔文字の感情分類, 言語処理学会　第 21 回年次大会　発表論文集, pp. 441–444 (2015).

[8] 中野滋徳, 足立顕, 牧野武則: 提題表現に基づく重要段落抽出, 情報処理学会研究報告. NL, 自然言語処理研究会報告, Vol. 162, pp. 159–166 (2004).

[9] 佐藤元紀, 伊藤孝行: Paragraph Vector と多層パーセプトロンを用いた有害文書の分類手法, 情報処理学会第 77 回全国大会, pp. 165–166 (2015).

[10] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank, *Conference on Empirical Methods in Natural Language Processing*, pp. 1–12 (2013).

[11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.:Scikit-learn: Machine Learning in Python, *The Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830 (2011).