深層学習による文章の分散表現を用いた落語の会話の一貫性推定

1　はじめに

近年, 機械学習の発展が目覚ましく, 自然言語の分野においても大きな成果を上げている. 特に叙述的な文章に関しては, 文章の理解にとどまらず, 文章の生成もなされており, 研究を超えて実用化に至っているものも多くある. 一方で口述的な文章は, 叙述的な文章よりも必要な要素と必要でない要素が入り混じるため複雑であり, 研究があまり進んでいない. そのなかでも, 喜劇や落語といったものは普通とは異なる言動が多くみられる一方で, それらの行動及び会話が支離滅裂にならないようにする必要があるため, 一般的な文章に比べ, 機械学習による文章の理解は非常に困難であるとされる.

以上の点を背景として, 本実験では落語に着目した. 落語は, その中にほぼ必ずオチを含み, 1 つの物語の中で, 登場する人物らが語り合うなかで物語が進んでいく作品が多く, 古い言葉を用いることが多々ある. そのため, 機械学習による理解が困難なものの一つに挙げられる. そこで, 本実験では, 機械学習による落語の理解の前段階として落語における会話文の一貫性の推定をすることを目的とする.

2　要素技術

2.1　ニューラルネットワーク

ニューラルネットワーク (Neural Network, NN)[1] は, 人間の神経細胞から着想を得た脳神経系のニューロンを数理モデル化した工学的手法である. 本実験では, NN のうちの, 階層型ニューラルネットワークである多層パーセプトロン (Multi-Layer Perceptron, MLP) を使用した. MLP は, 神経細胞をモデル化した多数のノードを層状に結合した階層構造として表される. 図 1 に簡略図を示す. 入力側に配置されたノードの層を入力層, 出力側に配置されたノードの層を出力層といい, その間の層を中間層という. MLP が構成するノードはニューロンと呼ばれる. (1) 式にそれぞれのニューロンの出力を示す. (1) 式において, $\boldsymbol{w_o}$ はバイアス, $\boldsymbol{w_i}$ は重み, $\boldsymbol{x_i}$ は入力データを意味する. f は活性化関数を表し, この関数によって非線形関数となる各ノードは n 個の入力と 1 つの出力を持ち, 前の層の出力を入力として受け取り, 出力することを繰り返すことで出力層で出力する. それぞれのノードの適切なパラメータを学習して求めることにより, 最適な出力を導き出す. 本実験では 5 層の MLP を用いた.

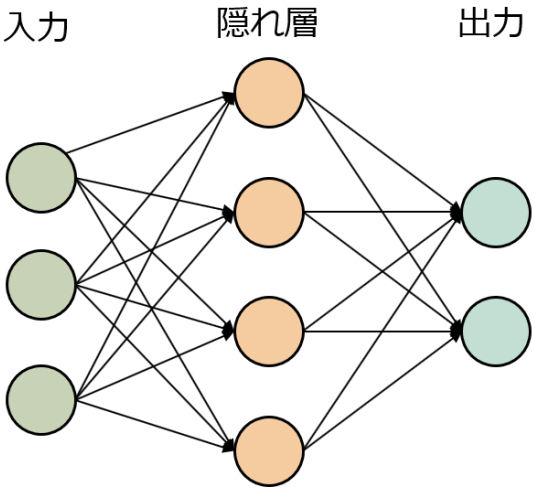

図 1: MLP モデルの簡略図

$$\boldsymbol{y} = f(\boldsymbol{w_o} + \sum_{i=1}^{n} \boldsymbol{w_i} \cdot x_i) \tag{1}$$

2.2　Word2Vec

Word2Vec[2] は, ニューラルネットワークを用いて単語の分散表現を獲得する手法である. 同じ文脈で出現する単語は類似した意味を持つと予想されることに基づき, テキスト中の各単語を周辺の単語から予測する擬似的な単語予測のタスクを設定し, このタスクを大量のテキストからニューラルネットワークで学習し, 中間層における各単語の重みを抽出することによって, 単語に対する概念ベクトルを獲得する. 単語を分散表現で扱うことにより, 単語を意味空間上の各点に対応させることができ, 単語間の意味を考慮した類似度の測定や, 意味的な計算が可能となる. ある単語の周辺の文脈から中心の単語を推定するモデルを Continuous Bag of Words (CBoW) といい, 中心の単語から文脈を推定するモデルを Skip-gram という.

2.3 Doc2Vec

Doc2Vec[2] は, Word2vec を文書単位へと拡張した手法である. 上記の CBoW を拡張したモデルを Paragraph Vector with Distributed Memory (PV-DM) といい, Skip-gram を拡張したモデルを Paragraph Vector with Distributed Bag of Words (PV-DBoW) という. 今回の実験では, PV-DM を用いた. Doc2Vec は分散表現として単語ベクトルとパラグラフベクトルを保持し, これらはニューラルネットワークの中間層の重みで表現される. パラグラフは局所的な文脈の中で失われた情報を代表するものであり, パラグラフのトピック等抽象度の高い情報を記憶する機能を持つ. Doc2Vec は, 文書の感情極性判定や文書検索の評価実験において, 長文, 短文に関わらず, 単語の語順を考慮しない Bag-of-Words などのそれまでの手法を大幅に上回ることが報告されている.

2.4 形態素解析

日本語は, 英語とは異なり, 文中で単語が区切りなく並べられている. そのため, Doc2Vec を用いる前に, 文を単語に区切る必要がある. 文を単語に区切りその品詞, 読み, 活用などを解析することを形態素解析といい, 日本語や韓国語で自然言語を扱う際の大きな問題の一つとなる. そして, JUMAN++[3] は京都大学により配布されている日本語文の形態素解析をするソフトウェアである. 従来の形態素解析は, 局所的な文法的制約や, 数万文程度の訓練データ中の局所的な単語並びの傾向を学習したものであったため, 広い文脈での意味的整合性は考慮していなかったが, Recurrent Neural Network (RNN) に基づく言語モデルを利用することにより, 文全体として意味的に妥当な単語分割が実現されている.

3 実験

今回の実験の目的は, 落語の文章から Doc2Vec を用いて文章の分散表現を獲得し, それを用いて落語の会話文の一貫性を推定する事である.

3.1 実験手法

3.1.1 分散表現の獲得

落語の文章を訓練データとテストデータに分け, 訓練データの文章を Doc2Vec に学習させた. 表 2 に Doc2Vec の学習時のパラメータを示す. 学習したモデルにそれぞれの文を入力し, それぞれの文の分散表現を獲得した.

3.1.2 MLP による分類

すべての作品のなかの, 連続する 5 文の会話の集合を正順群とした. また, 正順でない 5 文の会話の集合として, 同作品中のランダム箇所から 5 つの文をまとめた群 (random 群), 正順の 5 文をバラバラに入れ替えた群 (shuffule 群), 正順の最後の一文を同作品中のランダムな会話文に入れ替えた群 (replace 群), 正順の 5 文を逆順に入れ替えた群 (reverse 群), をそれぞれすべての作品の中から正順である群の要素数と同数用意した. 表 1 にそれぞれの群の例を示す. 正順群ともう一方の群を混合した集合をそれぞれ作った. そして, そのなかの要素に対し, , すべての文章をベクトル化し, それを 5 文の順番につなげたものを入力として, そのベクトルが正順である群とそうでないもう一つの群かどうかを推定器によって推定した. 推定器として MLP を用いて, これを訓練データによって学習した. どの集合に対しても, MLP のハイパーパラメータはそろえた. 表 3 に MLP の学習時のパラメータを示した.

3.1.3 推定器の評価

訓練データによって学習した MLP を, テストデータの 5 文の会話が正順群かそうでないもう一方の群かを推定することにより検証した. その精度について正解率 (Accuracy), 適合率 (Precision), 再現率 (Recall), F 値 (F) を求めることで評価した. また, 5 文を作品での進行度によって分け, それぞれの進行度ごとの正解率をまとめた. 同様に, 5 文の中の未知語彙の個数によっても分け, それぞれの個数ごとの正解率をまとめた.

3.2 使用データ

今回, 落語のデータとして青空文庫にある三遊亭円朝と鈴木 行三の作品を利用した. それぞれの作品

表 1: 5 会話文の群の名前と例

群	例 (index)
正順	1, 2, 3, 4, 5
random	2, 7, 10, 1, 8
shuffule	4, 2, 3, 5, 1
replace	1, 2, 3, 4, 7
reverse	5, 4, 3, 2, 1

表 2: Doc2Vec のパラメータ

パラメータ	値
学習モデル	PV-DM
埋め込みサイズ	20
ウインドウサイズ	4
学習率	0. 05

表 3: MLP のパラメータ

パラメータ	値
入力層の次元数	100
隠れ層 1 のノード数	50
隠れ層 2 のノード数	25
隠れ層 3 のノード数	12
隠れ層 4 のノード数	6
出力層の次元数	2
バッチサイズ	128
学習率	0. 0001
活性化関数 (隠れ層)	relu
活性化関数 (出力層)	softmax
最適化手法	Adam
目的関数	categorical cross entropy
最大エポック数	1000

から余分な言葉を消し, JUMAN++ を用いて単語ごとに空白で分けたものを実験に用いた. 表 4 に訓練データおよびテストデータのそれぞれのパラメータを示す.

表 4: 使用データ

	パラメータ	個数
train	作品数	43
	連続する 5 文の会話数	1047
	語彙数	10572
	総単語数	95997
test	作品数	3
	連続する 5 文の会話数	129
	語彙数	1751
	総単語数	6735
	未知語彙数	496

4 実験結果

表 5 にテストデータを識別したときの識別率を示す. 全体としてみると, 正順とどの群との対比においても, 訓練データに対しては正解率, 適合率, 再現率, そして F 値のすべての値に対して高い識別率を得られた一方で, テストデータに対しては, 群ごとに差がみられたが, その差は小さい範囲に収まった. 特にどの群との対比においても F 値のベースラインである 0.6666 を上回ることはできなかった. また, 4 つの識別の結果を合わせ, 5 文の物語上での出現箇所で分け, その場所ごとに, すべての識別の中で正しく識別した個数の割合をとった. 図 2 にその結果を示す. 同様に 4 つの識別の結果を合わせ, 5 文の中の未知単語の個数で分け, その個数ごとに, すべての識別の中で正しく識別した個数の割合をとった. 図 3 にその結果を示す.

5 考察

図 2 を見ると, 話が 3 割から 5 割進んだところで正答率が上がり, 最後の少し前にもう一度上がり再び下がっていることがわかる. 初めの正答率が上がるのは, 落語のストーリーが起承転結に基づいており, 承の部分ではあまり話が飛躍せず, 一貫性がわかりやすくなったことが原因だと考えられる. また, 最後の少し前に一度正答率が上がっているのはオチの前に話の一貫性が分かりにくくなると, オチの面白さが少なくなってしまうために, オチの前には話の流れがいったんわかりやすくなり, そこでオチがきて, 一貫性のわかりにくい文章になったことが原因だと考えられる.

図 3 から, 未知の単語の数が増えても正答率は下がっていないため, 未知の単語の数は, 今回の精度に大きく関与していないことが考えられる. そのため, あまり精度が出なかった原因として, まず, 落語中の会話の中には古い言葉が多くあり形態素解析が

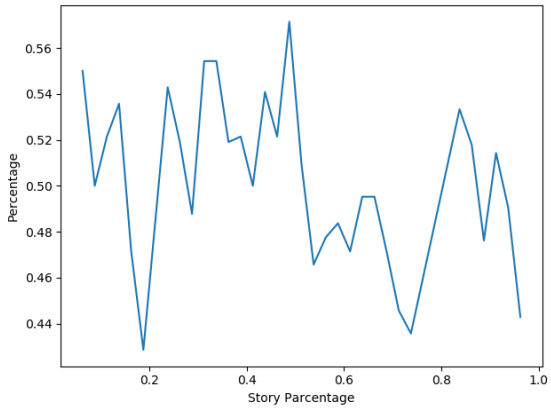

図 2: 出現箇所と正答率

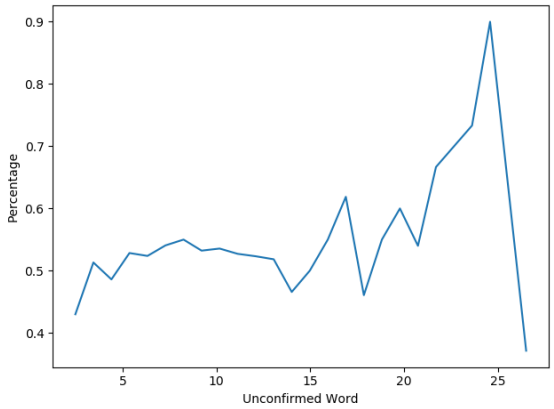

図 3: 未知語彙数と正答率

うまくいかなかったことが考えられる.

もう一つの原因として, あいさつや返事といった会話文は, それぞれがよく似ており, それらを入れ替えた正順の文と非常によく似ている 5 文が非正順群に入ったことが考えられる. そして, 逆に 1 文がとても長文のものがあり, それによって文の意味をうまく分散表現にできなかったことも考えられる.

それぞれの群を比較すると, random が最も高い正解率となった. これより, 5 つの文章がつながっていることを区別する何らかの判断基準が文中にあることが考えられる. その一方で, replace は F 値が非常に低かった. そのため, 落語の会話文が 5 文にわたって一貫しているかどうかはうまく推定できていないことがわかる. replace が低かったことの原因として, 5 文目が別のものと変わってもそこまでおかしくない文章であった場合, 正順と同じような 5 文が replace として扱われたことが考えられる.

表 5: それぞれの識別の精度 (テスト)

比べた群	正解率	適合率	再現率	F 値
random	0.5724	0.5826	0.5108	0.5444
shuffule	0.5034	0.5027	0.6410	0.5635
replace	0.5023	0.5045	0.2558	0.3395
reverse	0.5027	0.5036	0.3767	0.4310
ベースライン	0.5000	0.5000	1.0000	0.6667

6 まとめと今後の課題

本実験では, 機械学習による落語の理解の前段階として落語における会話文の一貫性の推定をした. 結果として, 起承転結の承の部分と, オチの直前には文章の一貫性が保たれやすいことが分かった. 一方で文章すべてにおいて一貫性があるかどうかを推定することはできなかった. 今後の課題として, 大規模コーパスを用いて Doc2Vec に学習させ, それを用いて文章のベクトル化を行うことがあげられる. また, Doc2Vec, 及び MLP の最適なパラメータの調整をすることも課題としてあげられる. そして, 落語以外の様々な種類の文章においてその違いを検証することも挙げられる.

参考文献

[1] 萩原克幸. ニューラルネットワークの基礎と理論的に重要な課題. プラズマ・核融合学会誌= Journal of plasma and fusion research, Vol. 82, No. 5, pp. 282–286, 2006.

[2] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, Vol. abs/1405.4053, , 2014.

[3] 京都大学大学院情報学研究科黒橋・河原研究室. 日本語形態素解析システム juman++ version 1.0. 2016.