

修士学位論文

題 目

分散表現化手法を用いた4コマ漫画の
文脈に基づく順序推定

主査 森直樹 教授

副査 藤本典幸 教授

副査 吉岡理文 教授

令和元年(2019年)度修了

(No. 2180104008) 岩崎凌

大阪府立大学大学院工学研究科
電気・情報系専攻 知能情報工学分野

分散表現化手法を用いた 4 コマ漫画の文脈に基づく順序推定

Context-based Frame Ordering Recognition of Four-Scene Comics

using Distributed Representations

分 野 名	知能情報工学分野	氏 名	岩崎 凌
Department	Computer Science and Intelligent Systems	Name	Ryo IWASAKI

Recently, in the field of artificial intelligence (AI), novels, music and comics have been researched. Computers understanding arts and creating automatically are very significant challenges.

A task, Frame Ordering Recognition (FOR), is used in the field because data for the task are easy to obtain. In the field, as long as existing datasets, which should be used to avoid copyrights, have lines, we use techniques in natural language processing (NLP).

Coordinates information is so effective that AI solves the task easily. However, AI cannot obtain perfect scores. To infer order of some episodes, computers have to use contextual information. Thus, a model which understands contexts is needed.

Although comics have linguistic information and visual information, and we can also solve the task with computer vision (CV) or multi-modal techniques, I solved it with NLP techniques. This is because we should extend our knowledge of FOR with NLP and CV before multi-modal analysis.

In this research, I used “OL lunch”, a four-scene comic in Manga 109. I thought four-scene comics were easy to predict order. Four-scene comics have stories which are along Ki-Sho-Ten-Ketsu (起承転結), the structure and development of Japanese narratives. Also, contexts in four-scene comics are closed in four frames.

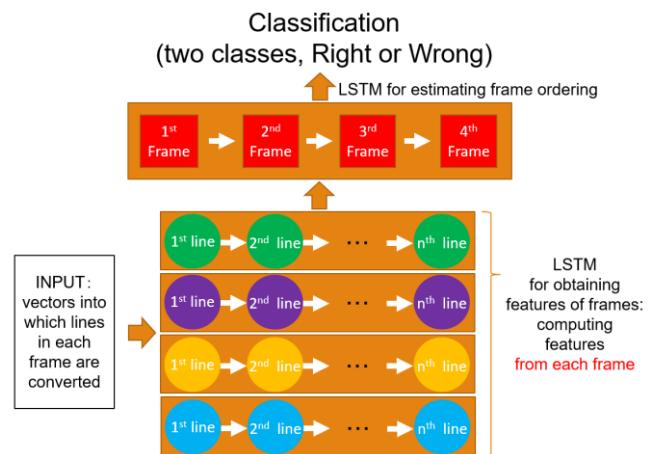


Figure.1: Outline of the proposed model

I proposed a model for FOR. Figure.1 shows the outline of the proposed model, which is consisted of two long short-term memories (LSTMs). Inputs into the model are vectors converted from lines. To convert lines, I used three models, Doc2Vec, Skip-Thought and Bidirectional Encoder Representations from Transformers (BERT). I sorted four-scene comics randomly and predicted whether the sorted comics are in right order or wrong order.

As a result, in comparison with four-scene comics in the right order (order: 1234) and in the wrong order (order: 2431), the model with vectors from BERT achieved 69.8 % on test data. Also, in both comparison with comics in right order and in wrong order (order: 2143, 4312 and so on), which is very different from right order and comparison with comics in all wrong order, the model outperformed baseline-scores.

分散表現化手法を用いた 4 コマ漫画の文脈に基づく順序推定

第 1 グループ 岩崎 凌

1. はじめに

近年、人工知能による小説や音楽、アニメ、漫画といった創作作物を対象とした研究が大きな関心を集めている。創作作物理解や作品の自動生成といった試みは工学的に興味深く意義が大きい反面、そもそも人の創作物理解は高次の知的活動であり、どういったタスクであれば計算機が創作物を理解したといえるのかを定義することさえ現状では難しい。

漫画を対象とした研究はコピーライトの問題で十分なデータが手に入らないなどの多くの制約により、キャラクター識別やセリフや吹き出し識別、コマの順序推定といった現状のデータでも可能なタスクを設定した研究が一般的である。

この中で自然言語の情報を用いた、あるいは、用いることができると言えられるタスクはコマの順序推定である。順序推定がタスクとして設定されやすい理由は 3 つある。順序ラベルのアノテーションが誰でも可能であること、電子コミックの登場によりコマの順序というメタ情報が商業的にも利用できるというモチベーションがあること、コマの座標情報を用いることで比較的簡単に解くことができるタスクであることである。

コマの順序推定における座標情報は重要な情報であり、非常に高い精度でコマの順序を推定できることが知られている。ただし、4 コマ漫画や 4 コマ漫画ではない漫画における普遍的なコマ割りなどと異なり、特殊なコマ割りをしている場合には、座標情報のみで解くことは困難であり、コマ間の文脈を理解したモデルの構築が必要不可欠となる。

漫画は文字と絵によって情報伝達を行うため、コマの順序推定を解くにあたり、セリフを対象とした文字情報によるアプローチ、絵を対象とした視覚情報によるアプローチ、あるいは、それら両方を用いたマルチモーダルなアプローチを取ることができる。漫画という媒体の情報をフル活用するにはマルチモーダルでの解析が最もよいと考えられるが、マルチモーダルでの解析には自然言語処理による解析と画像処理による解析のそれぞれの十分な見知が必要不可欠である。したがって、本研究は座標情報を用いないマルチモーダルなコマの順序推定の第一歩という位置づけのもと、自然言語を用いた解析を行う。

2. 実験データ

2.1. データセットについて

漫画の研究では、コピーライトの問題を回避するために既存のデータセットを使用することが望ましい。既存のデータセットとして、4 コマ漫画ストーリーデータセットや eBDtheque, Manga 109 [1][2] などが存在する。4 コマ漫画ストーリーデータセットは他のデータセットに比べ、ストーリー解析のための様々なメタ情報が含まれるが、データ数が少ないという問題がある。eBDtheque は日本語のみならず、英語などの多言語で書かれた漫画を持っているが、本実験では日本語での解析のみを目的としているうえ、4 コマ漫画ストーリーデータセットと同様にデータ数の問題がある。Manga 109 は、日本語の漫画 109 冊により構成されるデータセットであり、上記の 2 つのデータセットに比べてデータ数が多く、また座標やテキストといった順序推定をするためのメタ情報も含まれる。したがって本研究には Manga 109 を用いることとし、その中でも「OL ランチ」をデータとして用いた。ノイズになる可能性を考え、複数の漫画は用いなかった。

2.2. データについて

データとして使用する「OL ランチ」のジャンルは 4 コマ漫画である。4 コマ漫画は座標情報によって最も順序を推定しやすい漫画である。また、4 コマ漫画というジャンルの漫画が必ずしも 4 コマ漫画からなるわけではなく 5 コマや 8 コマからなる漫画も存在するものの、「OL ランチ」内のエピソードはすべて 4 コマからなる。したがって、今回使用するデータは文脈を加味する必要はなく、座標情報によって容易に推定可能である。そのうえで、4 コマ漫画をデータとして用いる理由は、今回の実験の目的がコマ間の文脈によって順序を推定するモデルの構築であるため、4 コマで文脈が完結する可能性が高く（文脈を理解するために一体いくつのコマを把握しておく必要があるかを考える必要がなく）、起承転結からなるストーリーの流れが理解しやすい 4 コマ漫画こそ文脈を理解して順序を推定するモデルの構築の第一歩にふさわしいデータと考えたためである。

2.3. 実験難度

実験前に、セリフを有するコマ数の観点から自然言語処理手法を用いてアプローチする場合の実験タスクの難易度について考察する。

比較的解きやすい問題として 4 コマ漫画を選択したが、問題としての難易度は低くない。自然言語処理の場合、セリフが存在しないデータの予測はできない（これらのデータは今回の自然言語処理手法のみを用いる場合には絶対に正しく識別することができないが、今後画像やマルチモーダルから順序を推定することを考え、比較のために使用した）。

セリフを有するコマが 2 個以下の場合は解くことができないことは明白である。3 個の場合は不可能ではないが、困難な問題である。すべてのコマにセリフを有する例とセリフを持たないコマが 1 つある例の場合の例を挙げておく。以下に提示する順番はランダムに並び替えられており、実際に計算機が解く問題を体験することができる。

例 1

1. ガキと動物か
男は男で「かわいい」の範囲狭すぎないか
2. コレ？ チョーかわいいでしょ
あゆっぽいでしょ
3. あかわいい
えっ誰？
4. コギャルにとていいものの形容詞は「かわいい」しかないんか
コートがかわいいのか

例 2

1. これで若者とシンボクがはかれると思ってんだもんね
ま 1 年に 1 度だしガマンガマン
2. 楽しそうに振るまわなきゃ!!
オヤジなりに気苦労があるので
3. とはいえあたしだって行きたくない
楽しんでんのはオヤジだけだよね
ほかに旅行行かないし

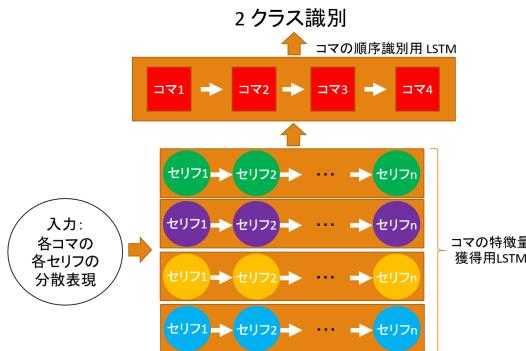


図 1: 提案モデル

4. (セリフ無し)

正解は、例 1 は 2 4 3 1、例 2 は 3 1 4 2 である。これらの例が 4 コマ漫画から抜粋されていることを考えれば、起承転結になっているあるいはなっている可能性が高いことがわかり、物語の流れが判然とではあるが見えてくる。人間であれば、セリフを持たないコマが 1 つある場合には、物語の流れを理解した上でどこにセリフを持たないコマを置くと物語が自然になるかを考えることになり、問題がより難しくなっていることがわかる。なお、セリフを持たないコマは、例 2 では中年の上司が部下の文句を聞いてしまったコマである。

セリフを持たないコマが何コマ目に存在しやすいのか傾向を調べると、「OL ランチ」では、例 2 のような 3 コマ目に存在する場合に加え、1 コマ目にもセリフを持たないコマが集中することが統計的にわかる。最も出現回数の多い 3 コマ目でも 45 個中 23 個と約 50 % という偏り具合であり、統計的に問題を解く(つまり文脈を加味せず確率的にもっともらしい 3 コマ目をセリフを持たないコマとする)場合には、半数近くを間違えることとなる。したがって、計算機も人間と同様に、どこにセリフを持たないコマを置くと物語が自然になるかを文脈的に考える必要がある。計算機がこういったデータをうまく識別できるかどうかを確認することも今回の実験のポイントとなる。

3. 数値実験

3.1. モデル

図 1 にコマの提案モデルの概要を示す。以下、このモデルを識別用 LSTM モデルと呼ぶ。このモデルは、コマの特徴量獲得用 LSTM とコマの順序推定用 LSTM の 2 つの LSTM からなる。まず 1 コマ目のセリフ、2 コマ目のセリフ、3 コマ目のセリフ、4 コマ目のセリフをそれぞれ LSTM に入力し、それぞれのコマの特徴量を獲得する。これらのセリフはあらかじめ Doc2Vec, Skip-Thought および BERT で分散表現化しておいた。また、各コマのセリフの分散表現を入力する際、頭に零ベクトルを入力した。その後、あらかじめ並び替えられたコマの順番通り、順序を推定するための LSTM と全結合層に入力して正例・負例の 2 クラス識別をした。なお、コマ外にはみ出ているセリフや一部の書き文字は用いなかった。コマ外に出ているセリフを使わなかつたのは、コマ外セリフの所属判定が機械的に困難であるためであり、書き文字の一部を用いなかつたのは、そもそも Manga 109 のデータが書き文字をデータとして持つていなかつたためである。

3.2. 実験

全 208 エピソードのうち、ランダムで 50 エピソードをテストデータとし、残りを訓練データとした。実験 1 と同様に「OL ランチ」の各エピソードのコマを(正例 1 つを含む)24 通りに並び替え、入力された順番が正例か負例かを識別するので、正例と負例が 1 : 23 の比率で存在する不均衡データで

あることに注意する。過学習をしていないモデルの結果を提示するために、10 分割交差検定を行い、評価データが最も良い Epoch で評価した。

また、学習したモデルを用いて、正しい順序の 4 コマ漫画と誤った順序の 4 コマ漫画から正例を選択できるかどうかを確認した。

3.3. 実験結果と考察

モデルの識別層から正例である確率を獲得し、これらを比較した。最も高い確率で正例だと判断した順序をモデルの答えとし、正しい順序や間違った順序に並び替えられた 4 コマ漫画の中から正解を選択できるかどうかをいくつかのパターンでの比較で調べた。

1. 正しい順序と間違った各順序の確率を 1:1 で比較。ベースラインは 50.0 %

- 最も正解率の高かった BERT の分散表現を用いたモデルでは、2431 との比較が最も正解率がよく 69.8 %, 3214 との比較が最も難しく 51.2 %, 平均で 58.8 % であった。

2. 正しい順序と各コマの順序が完全不一致な 9 通りの順序(2143 や 4312 など)の比較。ベースラインは 10.0 %

- BERT の分散表現を用いたモデルは 25.4 % のデータを正しく識別できた

3. 全 24 通りの比較。ベースラインは 4.17 %

- BERT の分散表現を用いたモデルの正解率は 9.20 %

パターン 1 での比較で最も識別が困難だったのは 3214 であったが、1 コマ目と 3 コマ目にセリフを持たないコマが集中することから、その点において統計的に類似した 1, 3 コマが難しかったのだろうと考えられる。また、計算機による順序推定では、各コマのセリフの数や長さ(アルゴリズムによってはセリフの長さが影響の大きい分散表現になっている場合が見られる)といった点から統計的に加味しながら解くため、人間の考える推定のしやすい問題とは多少の差異があり、また、統計的な観点から解きやすい順序と解きにくい順序が存在すると考えられる。

言語情報のみを用いて、日本の 4 コマ漫画の順序推定をした実験の例はなく、実験結果の評価は非常に難しいものではあるが、パターン 1, 2, 3 すべてにおいてベースラインを上回る識別率を得ることができた。

4. まとめと今後の展望

座標情報だけでは順序推定が難しい漫画のために、文脈から順序を推定するモデルを提案した。モデルの評価は、正しい順序と誤った順序の 4 コマ漫画の中からを選択できるかどうかとし、正しい順序と間違った順序 1 つずつを比較した場合には、最大で 69.8 % の精度で識別することができた。

今後の展望としては、現在のモデルを更に改良し、言語情報のみでの識別精度を上げることや、画像処理や言語と画像を用いたマルチモーダルなモデル、最終目標である座標情報だけでは推定できない漫画のための座標情報と組み合わせたモデルの構築が挙げられる。

参考文献

- [1] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017.
- [2] T. Ogawa, A. Otsubo, R. Narita, Y. Matsui, T. Yamasaki, and K. Aizawa. Object detection for comics using manga109 annotations. *CoRR*, abs/1803.08670, 2018.

目次

1 はじめに	1
2 要素技術	3
2.1 分散表現化手法およびその関連技術	3
2.1.1 Word2Vec	3
2.1.2 Doc2Vec	5
2.1.3 Skip-Thought	7
2.1.4 Transformer	8
2.1.5 BERT	12
3 実験準備	14
3.1 データセットについて	14
3.2 順序推定におけるデータについて	15
3.3 順序推定用モデル	17
3.4 順序推定の実験難度についての考察	17
4 実験 1	25
4.1 実験 1.1	25
4.1.1 実験 1.1 のデータについての注意	25
4.1.2 実験 1.1 結果 (数値)	26
4.2 実験 1.2	28
4.2.1 実験 1.2 のデータについての注意	28
4.2.2 実験 1.2 結果 (数値)	29
4.2.3 実験 1.2 結果 (4 コマ正解数)	30
4.3 実験 1.3	38
4.3.1 実験 1.3 のデータについての注意	38
4.3.2 実験 1.3 結果 (数値)	39
4.3.3 実験 1.3 結果 (4 コマ正解数)	39
5 実験 2	44
5.1 実験 2 で用いるデータセット	44

5.2 データについて	46
5.3 モデル	46
5.4 実験 2 の結果と考察	47
6 まとめと今後の課題	50
謝辞	51
参考文献	52

図目次

2.1 CBOW 概要図(参照 : Efficient Estimation of Word Representations in Vector Space ^[1])	4
2.2 Skip-gram 概要図(参照 : Efficient Estimation of Word Representations in Vector Space ^[1])	4
2.3 DM 概要図(参照 : Distributed Representations of Sentences and Documents ^[2])	5
2.4 DBOW 概要図(参照 : Distributed Representations of Sentences and Documents ^[2])	6
2.5 Skip-Thought 概要図(参照 : Skip-Thought Vectors ^[3])	7
2.6 Transformer 概要図(参照 : Attention Is All You Need ^[4])	10
2.7 Scale Dot-Product Attention 概要図(参照 : Attention Is All You Need ^[4]) . . .	11
2.8 Multi-Head Attention 概要図(参照 : Attention Is All You Need ^[4])	11
2.9 BERT 概要図(参照 : BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding ^[5])	12
3.1 OL ランチ © さんりようこ(出典 : Manga 109)	16
3.2 識別用 LSTM モデル	18
3.3 例 1 © さんりようこ(出典 : Manga 109)	23
3.4 例 2 © さんりようこ(出典 : Manga 109)	24
4.1 BERT の分散表現	33
5.1 4 コマ漫画ストーリーデータセットのデータの一例	45
5.2 事前情報なしモデル	47
5.3 事前情報ありモデル	48

表 目 次

3.1 モデルのパラメータ	21
3.2 実験データにおけるセリフを有するコマ数	22
3.3 セリフを有するコマが 3 個の各コマ番号におけるセリフのないコマの個数	22
4.1 実験データ数	25
4.2 実験結果 (BERT)	26
4.3 実験結果 (Skip-Thought)	27
4.4 実験結果 (Doc2Vec)	27
4.5 実験データ数	28
4.6 実験結果 (BERT)	28
4.7 実験結果 (Skip-Thought)	29
4.8 実験結果 (Doc2Vec)	29
4.9 テストデータの正例選択率および選択数	30
4.10 「OL ランチ」の各コマの長さとセリフの数の平均と分散	32
4.11 3 コマ目一致および不一致と 4 コマ目一致および不一致の平均	33
4.12 4 コマ順序推定選択率 (一対比較)	35
4.13 4 コマ順序推定選択率 (不完全一致比較)	36
4.14 4 コマ順序推定選択率 (全通り比較)	37
4.15 実験データ数	38
4.16 実験結果 (BERT)	39
4.17 実験結果 (Skip-Thought)	39
4.18 実験結果 (Doc2Vec)	40
4.19 テストデータの正例選択率および選択数	40
4.20 4 コマ順序推定選択率 (一対比較)	41
4.21 4 コマ順序推定選択率 (不完全一致比較)	42
4.22 4 コマ順序推定選択率 (全通り比較)	43
5.1 実験 2 のデータ数	46
5.2 モデルのパラメータ	47
5.3 実験結果	49

1 はじめに

近年，人工知能による小説や音楽，アニメ，または漫画といった創作物を対象とした研究が大きな関心を集めている。創作物理解や自動生成といった試みは工学的に興味深く意義が大きい反面，そもそも人の創作物理解は高次の知的活動であり，どういったタスクであれば計算機が創作物を理解したといえるのかを定義することさえ現状では難しい。

漫画を対象とした研究はコピーライトの問題で十分なデータが手に入らないなどの多くの制約により，キャラクター識別^[6] やセリフや吹き出し識別^{[7][8]}，コマの順序推定^[9] といった現状のデータでも可能なタスクを設定した研究が一般的である。

この中で自然言語の情報を用いた，あるいは，用いることができると考えられるタスクはコマの順序推定である。順序推定がタスクとして設定されやすい理由は3つある。順序ラベルのアノテーションが誰でも可能であること，電子コミックの登場によりコマの順序というメタ情報が商業的にも利用できるというモチベーションがあること，コマの座標情報を用いることで比較的簡単に解くことができるタスクであることである。

コマの順序推定における座標情報は重要な情報であり，高い精度でコマの順序を識別できることが報告されている。ただし，4コマ漫画や4コマ漫画ではない漫画における普遍的なコマ割りなどと異なり，特殊なコマ割りの場合には，座標情報のみで解くことは困難であり，コマ間の文脈を理解したモデルの構築が必要不可欠となる。ただし，ここで文脈とは，文間やコマ間の論理的関係というような狭義の意味ではなく，言語情報や視覚情報といった物語の流れを多少なりとも把握できる情報とし，ストーリーを一切把握できない座標情報に対応する表現として用いた。

漫画は文字と絵によって情報伝達をするため，コマの順序推定を解くにあたり，セリフを対象とした文字情報によるアプローチ，絵を対象とした視覚情報によるアプローチ，あるいは，それら両方を用いたマルチモーダルなアプローチを取ることができる。漫画という媒体の情報をフル活用するにはマルチモーダルによる解析が最もよいと考えられるが，マルチモーダルでの解析には自然言語処理による解析と画像処理による解析のそれぞれの知見が必要である。

要不可欠である。したがって、座標情報を用いないマルチモーダルなコマの順序推定を目指し、本研究は自然言語を用いて解析をする。

以下に本研究の構成を示す。まず、2章では本研究で用いる要素技術について概説する。3章で本研究で用いるデータについて説明する。そして4章で実験手法とその考察を示す。また、順序推定のみならず、5章に、追加実験として行った漫画の登場キャラクターの感情分析結果を示す。6章で本研究の成果をまとめたうえで、今後の課題について述べる。

2 要素技術

本章では、本研究の提案手法に用いた技術について説明する。

2.1 分散表現化手法およびその関連技術

本研究では、自然言語処理およびマルチモーダルな解析に分散表現を用いる。分散表現は1986年から続く長い歴史があるとされており^[10]、近年では自然言語処理において必要不可欠な技術である。分散表現化手法の多くは、事前学習として教師なし学習をすることで、単語や文の汎用的なベクトルへの写像を得る。本研究のようなデータが限られていて、使用するデータから文法や意味的特徴を正しく学習できないという懸念がある場合には有効な手法であると考えられる。

2.1.1 Word2Vec

Word2Vec^[1]は単語の分散表現獲得手法である。Word2Vecによって写像したベクトルは性能が高く、単語間の意味を考慮した類似度測定や単語間の意味的加算・減算ができる。

Word2Vecは、ある単語に着目した際、周りの単語からその単語を予測する、あるいは、その単語から周りの単語を予測することにより分散表現を獲得する。前者によって学習する手法を特に Continuous Bag of Words (CBOW)といい、後者を Skip-gram という。図2.1、2.2にそれぞれの概要図を示す。

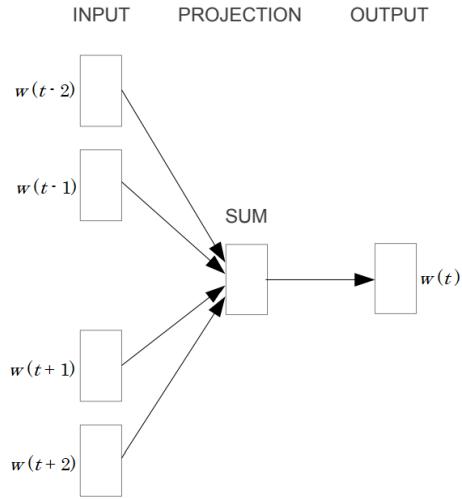


図 2.1: CBOW 概要図 (参照 : Efficient Estimation of Word Representations in Vector Space^[1])
ここで , $w(t)$ は , ある文における t 番目の単語である .

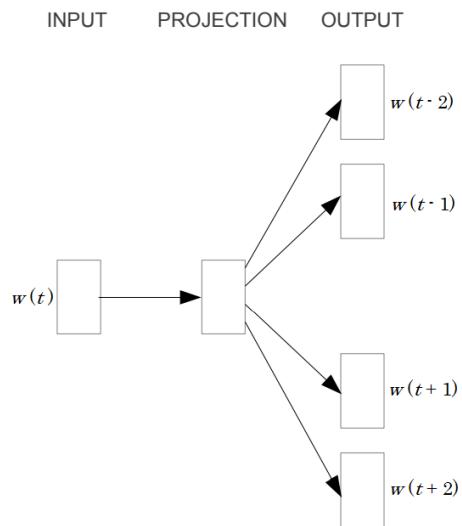


図 2.2: Skip-gram 概要図 (参照 : Efficient Estimation of Word Representations in Vector Space^[1])
ここで , $w(t)$ は , ある文における t 番目の単語である .

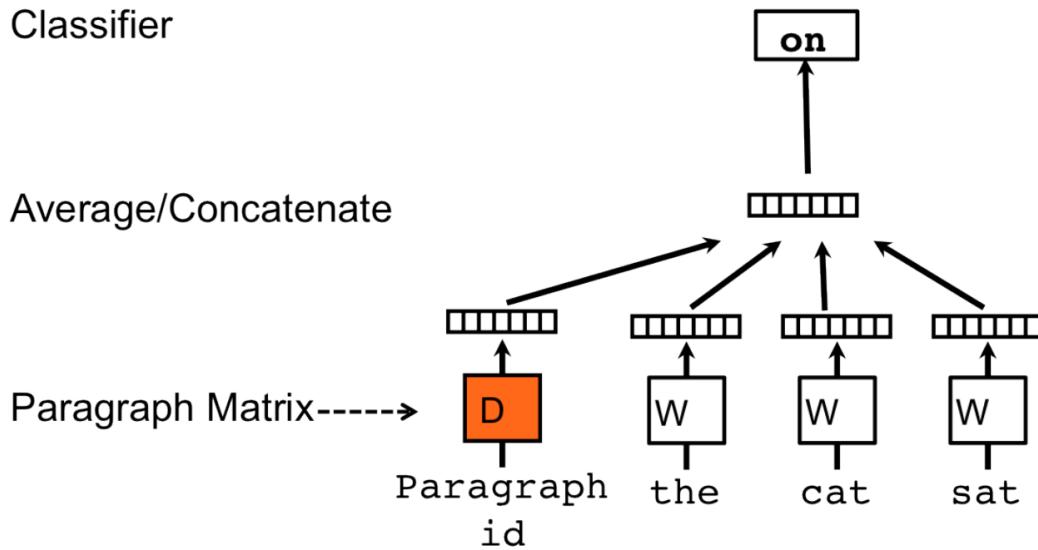


図 2.3: DM 概要図 (参照 : Distributed Representations of Sentences and Documents^[2])

2.1.2 Doc2Vec

Doc2Vec^[2] は Word2Vec から発展した , 単語および文の分散表現獲得手法である . Doc2Vec では文書を分散表現に変換するために Word2Vec に Paragraph ID を導入する . Paragraph ID は各文書と紐づいており , 単語の学習時に共に学習される . Doc2Vec では , この Paragraph ID を文の分散表現として見なす . Word2Vec の CBOW を拡張したモデルを Distributed Memory (DM) といい , Skip-gram を拡張したモデルを Distributed Bag-of-Words (DBOW) という . 図 2.3 , 2.4 にそれぞれの概要図を示す .

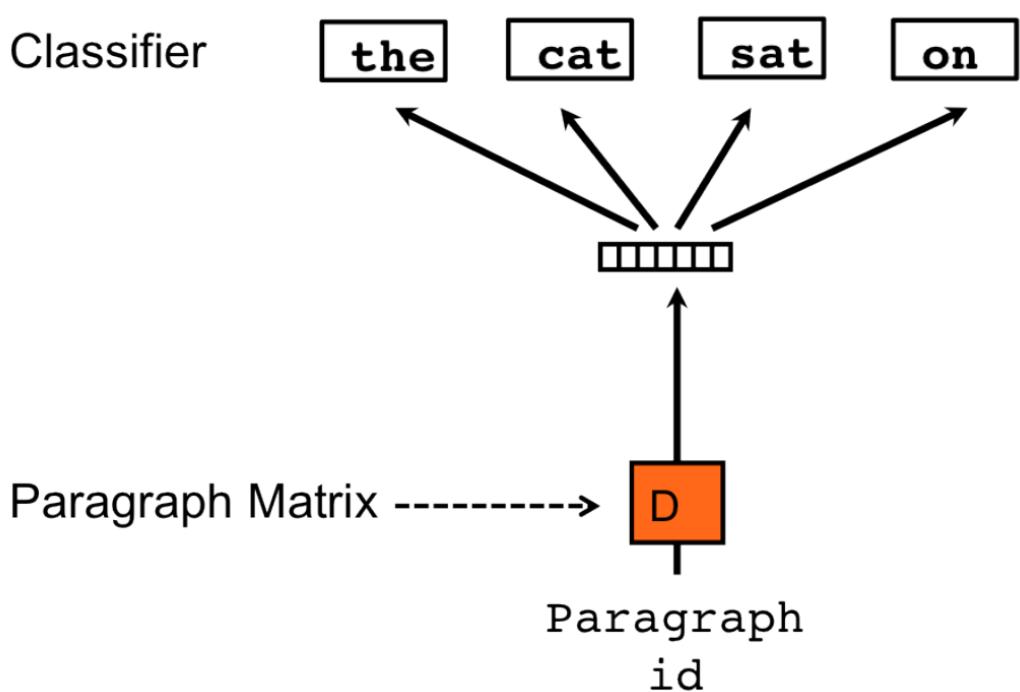


図 2.4: DBOW 概要図 (参照 : Distributed Representations of Sentences and Documents^[2])

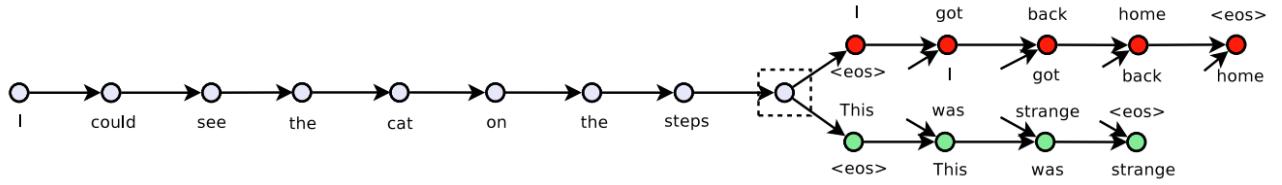


図 2.5: Skip-Thought 概要図(参照 : Skip-Thought Vectors^[3])

図は “I got back home. I could see the cat on the steps. This was strange.” という連続した文を用いて Skip-Thought を学習する場合の例である

2.1.3 Skip-Thought

Skip-Thought^[3] は教師なし学習による汎用的分散表現獲得手法である。図 2.5 に Skip-Thought の概要図を示す。

Skip-Thought は Encoder-Decoder モデル^[11]であり、文書内の文の連續性を学習する。具体的には、Skip-Thought の Encoder の入力をもとに、Decoder でその前後の文書を予測することで学習する。Encoder および Decoder には、Gated Recurrent Unit (GRU)^[12] や Long short-term memory (LSTM)^[13] のような Recurrent Neural Network (RNN) が用いられる。意味的特徴や構文的特徴が類似した文は似たベクトルに写像されることが期待される。

Skip-Thought は学習時に見られなかった単語に対する語彙拡張手法を有している。Word2Vec のような単語の分散表現獲得手法の学習済みモデルの単語ベクトル空間を \mathcal{V}_{w2v} とし、RNN の単語ベクトル空間を \mathcal{V}_{rnn} とする。このとき、 \mathcal{V}_{w2v} は \mathcal{V}_{rnn} に比べて大きいと考えられる。 $v \in \mathcal{V}_{w2v}$, $v' \in \mathcal{V}_{rnn}$ である v , v' に対し、 $v' = Wv$ すなわち $f: \mathcal{V}_{w2v} \rightarrow \mathcal{V}_{rnn}$ となるような W を求めることで単語を拡張する。

なお、本研究ではモデルの比較のため、使用できる単語をあらかじめ設定しているため、語彙拡張手法は使用されない。

2.1.4 Transformer

Transformer^[4] は RNN や Convolutional Neural Network (CNN) を用いず，Attention 機構^[14] のみで構成されたモデルである。図 2.6 に Transformer の概要図を示す。

RNN を用いないのは Transformer の設計思想として，並列化によって計算速度を向上させようという意図があるためである。これは RNN は時系列データに対して有効な手法であるものの，単語の位置に従い計算をするため，計算の並列化ができない，多くの計算時間がかかるという欠点があるからである。

また，Transformer は Encoder-Decoder モデルである。Encoder が， (x_1, \dots, x_n) で表現される入力シーケンスを連続値で構成されるベクトル $z = (z_1, \dots, z_n)$ へと写像し，そして同時に， z から出力 (y_1, \dots, y_m) を Decoder で出力する場合を考える。このとき，各ステップでそれ以前の出力を追加入力として考慮する。

Encoder 図 2.6 における N は，Encoder では $N = 6$ である。

各層は 2 つのサブレイヤーを持っている。1 つ目は Multi-Head Self-Attention 機構であり，2 つ目は単純な Position-Wise Feed-Forward Network からなる。

Position-Wise Feed-Forward Network を $\text{FFN}(x)$ とすると

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2.1)$$

となる。

Decoder 図 2.6 における N は，Decoder でも $N = 6$ である。Decoder では，Encoder における 2 つのサブレイヤーに加え，3 つめのサブレイヤーとして Multi-Head Attention を採用している。

Transfomer における **Attention** 機構 Attention は Query と出力のペアである Key と Value のセットからなる。ここで Query と Key と Value はすべてベクトルである。出力は Value の加重合計で計算される。このときの重みは Query と対応する Key から計算される値で各 Value と関係のある値である。

図 2.7 と 2.8 に Scale Dot-Product Attention と Multi-Head Attention を示す。

Scale Dot-Product Attention は d_k 次元の Query と Key と d_v 次元の Value を入力とする。Query とすべての Key のドット積を計算し、 $\sqrt{d_k}$ で各値を割り、ソフトマックス関数を適用することで Value に対する重みを得る。Scale Dot-Product Attention を $\text{Attention}(Q, K, V)$ とすると

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (2.2)$$

となる。なお、 V, K, Q はそれぞれ Value, Key, Query である。

Multi-Head Attention では異なる表現空間の情報を組み合わせて使用する。これは d_{model} 次元の Key と Value と Query を持つ 1 つの Attention 関数を使用するよりも、いくつかの線形写像を組み合わせたほうが効果的であるためである。Multi-Head Attention を $\text{MultiHead}(Q, K, V)$ とすると

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.3)$$

$$\text{ただし } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2.4)$$

となる。なお $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}} \times d_k}$ である。

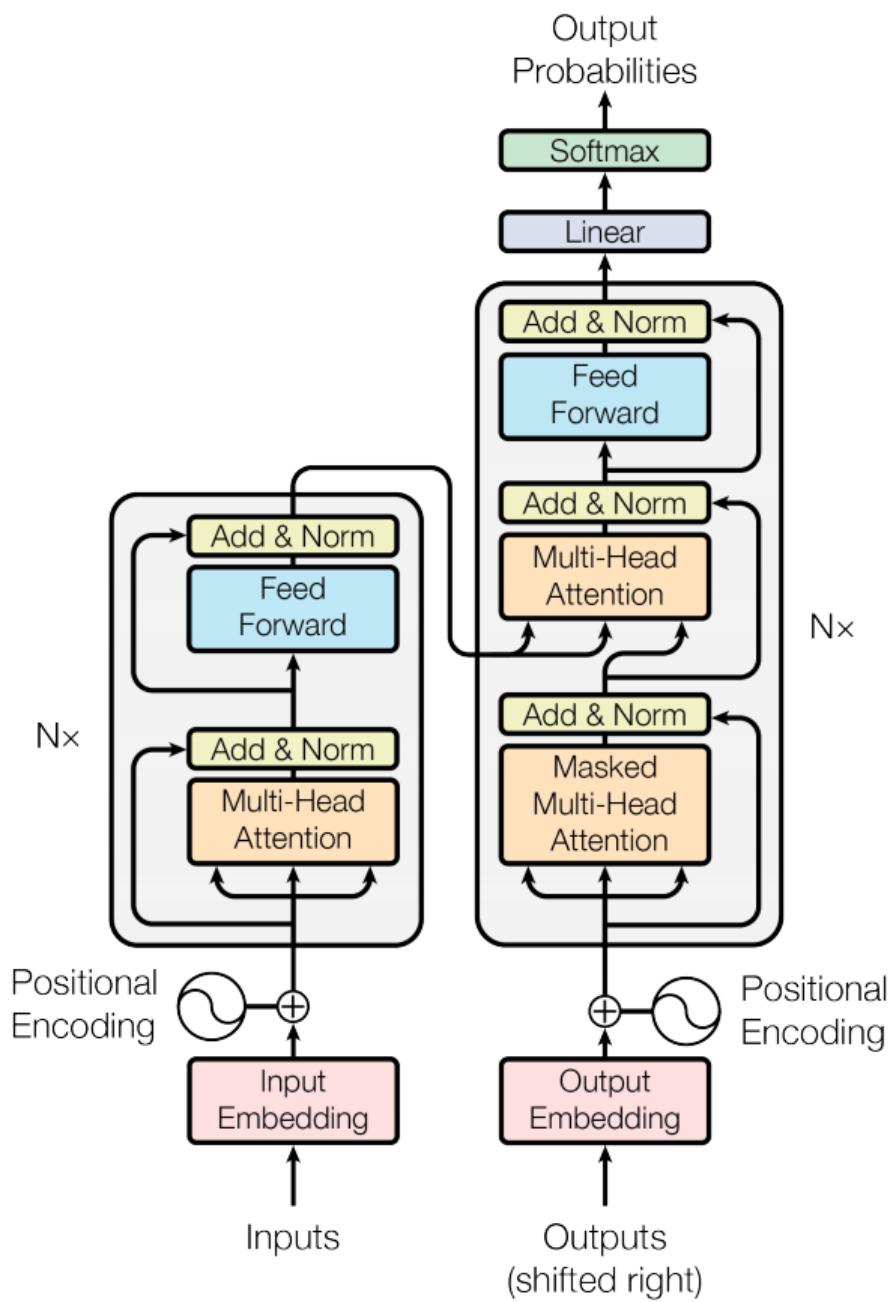


図 2.6: Transformer 概要図 (参照 : Attention Is All You Need^[4])

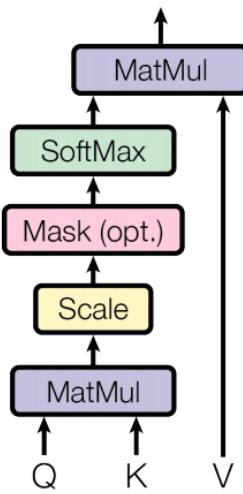


図 2.7: Scale Dot-Product Attention 概要図 (参照 : Attention Is All You Need^[4])
図における V, K, Q はそれぞれ Value, Key, Query である .

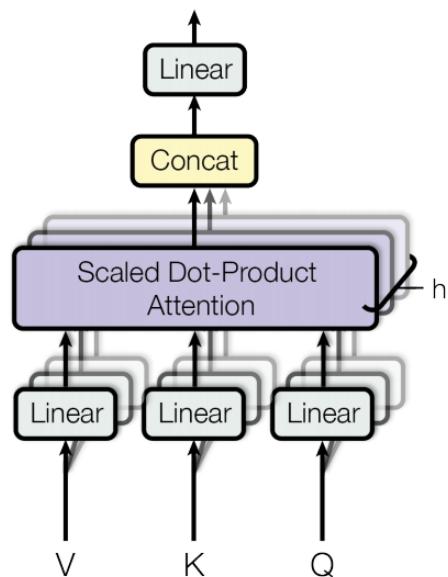


図 2.8: Multi-Head Attention 概要図 (参照 : Attention Is All You Need^[4])
図における V, K, Q はそれぞれ Value, Key, Query である .

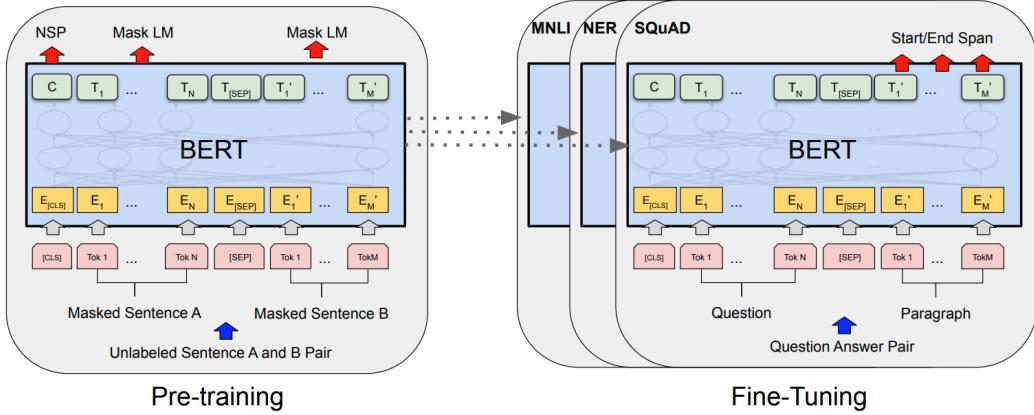


図 2.9: BERT 概要図 (参照 : BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding^[5])

BERT では Pre-training で学習したパラメータを Fine-tuning 時の初期値として使用する。Fine-tuning 時にはすべてのパラメータを学習する。[CLS] はすべての入力の先頭に置かれる特別なトークンで、[SEP] はセパレートトークンである。セパレートトークンは、例えば問題と答えという 2 文を入力とする際には、問題と答えの間に置かれる。

2.1.5 BERT

Bidirectional Encoder Representations from Transformers (BERT)^[5] はその名前の通り、Transformer^[4] によって構成される言語モデルである。図 2.9 に BERT の概要図を示す。BERT は ELMo (Embeddings from Language Models)^[15] をもとにしたモデルで、双方向 LSTM を Transformer に置き換えている。教師なし学習による事前学習をする際、Transformer によって、すべての層で左右の両方の文脈を加味した学習が可能となる。

BERT には事前学習と fine-tuning の二つの学習ステップが存在する。事前学習では、周囲の単語からある単語を予測する Masked Language Model (MLM) と 2 つ目の文章が 1 つ目の文章の次の文章であるかを予測する Next Sentence Prediction (NSP) によりモデルを学習する。MLM では、文章にマスクをかけた文章を予測する。上述の通り、通常の確率的言語モデルと異なり、左右どちらの文脈も加味した学習を実現する。

単語の分散表現化手法としては Word2Vec に加え、Glove^[16] や fastText

[¹⁷] [¹⁸] , ELMo などが存在するが , 本研究では単語の分散表現化手法として Word2Vec を用いた . これは , 本研究で単語の分散表現を使用する Skip-Thought の元論文が Word2Vec に基づいた内容であり , それに従ったためである .

3 実験準備

3.1 データセットについて

漫画の研究では、コピーライトの問題を回避するために既存のデータセットを使用することが望ましい。順序推定では、日本語の漫画 109 冊により構成されるデータセットである Manga 109^{[19][20]} を使用した。これはデータ数が多く、また座標やテキストといった順序推定をするために必要最低限なメタ情報を持っているためである。ノイズになる可能性を考え、複数の漫画は用いず、Manga109 の中でも「OL ランチ」をデータとして用いた。

なお、既存のデータセットとしては、Manga 109 の他に、4 コマ漫画ストーリーデータセット^[21] や eBDtheque^[22] が存在する。4 コマ漫画ストーリーデータセットは、人工知能を用いた漫画の研究のために作られたデータセットであり、研究者が漫画の執筆から関わっている。そのため、ストーリー解析のための様々なメタ情報が含まれているという特徴があり、様々なタスクで使用が見込まれる。eBDtheque は日本語のみならず、英語などの多言語にわたる漫画が含まれるという特徴がある。

3.2 順序推定におけるデータについて

図 3.1 に「OL ランチ」の 1 エピソードを示す。図 3.1 からわかる通り、OL ランチのジャンルは 4 コマ漫画である。

4 コマ漫画は座標情報によって最も識別しやすい漫画である。また、4 コマ漫画というジャンルの漫画が必ずしも 4 コマ漫画からなるわけではなく 5 コマや 8 コマからなる漫画も存在するものの、「OL ランチ」内のエピソードはすべて 4 コマからなる。したがって、今回使用するデータは文脈を加味する必要はなく、座標情報によって容易に推定可能である。そのうえで、「OL ランチ」をデータとして用いる理由は、今回の実験の目的がコマ間の文脈によって順序を識別するモデルの構築であるため、4 コマで文脈が完結する可能性が高く、つまり文脈を理解するために一体いくつのコマを把握しておく必要があるかを考える必要がなく、起承転結からなるストーリーの流れが理解しやすい 4 コマ漫画こそ文脈を理解して順序を推定するモデルの構築の第一歩にふさわしいデータと考えたためである。



図 3.1: OL ランチ © さんりようこ (出典 : Manga 109)

3.3 順序推定用モデル

図 3.2 にコマの順序識別用モデルの概要を示す。以下、このモデルを識別用 LSTM モデルと呼ぶ。このモデルは、コマの特徴量獲得用 LSTM とコマの順序識別用 LSTM の 2 つの LSTM からなる。まず 1 コマ目のセリフ、2 コマ目のセリフ、3 コマ目のセリフ、4 コマ目のセリフをそれぞれ LSTM に入力し、それぞれのコマの特徴量を獲得する。これらのセリフはあらかじめ Doc2Vec、Skip-Thought および BERT で分散表現化した。分散表現化に使ったこの 3 モデルのうち、BERT は pytorch transformers¹ と事前学習済みモデル^{2[23]} を使用した。Skip-Thought は自ら実装したモデルを使い、Doc2vec は gensim^{3[24]} を使った。これらはすべて日本語 Wikipedia⁴ を用いて事前学習している。なお、事前学習データ、訓練データ、テストデータはすべて Juman++^[25] と byte pair encoding^[26] によって分かち書きした。

また、各コマのセリフの分散表現を入力する際、頭に零ベクトルを入力した。その後、あらかじめ並び替えられたコマの順番通り、順序を推定するための LSTM と全結合層に入力して正例・負例の 2 クラスを識別した。なお、コマ外にはみ出ているセリフや一部の書き文字は用いなかったが、コマ外に出ているセリフを使わなかったのは、コマ外セリフの所属判定が機械的に困難であるためであり、書き文字の一部を用いなかったのは、そもそも Manga 109 のデータが書き文字をデータとして持っていないためである。表 3.1 に実験に使用したモデルのパラメータをまとめる。

3.4 順序推定の実験難度についての考察

実験前に、セリフを有するコマ数の観点から自然言語処理手法を用いた場合の実験タスクの難易度について考察する。

比較的解きやすい問題として 4 コマ漫画を選択したが、問題としての難易度は低くない。自然言語処理の場合、セリフが存在しないデータの予測はで

¹<https://github.com/huggingface/pytorch-transformers>

²<http://nlp.ist.i.kyoto-u.ac.jp/index.php>

³<https://radimrehurek.com/gensim/index.html>

⁴<https://ja.wikipedia.org/>

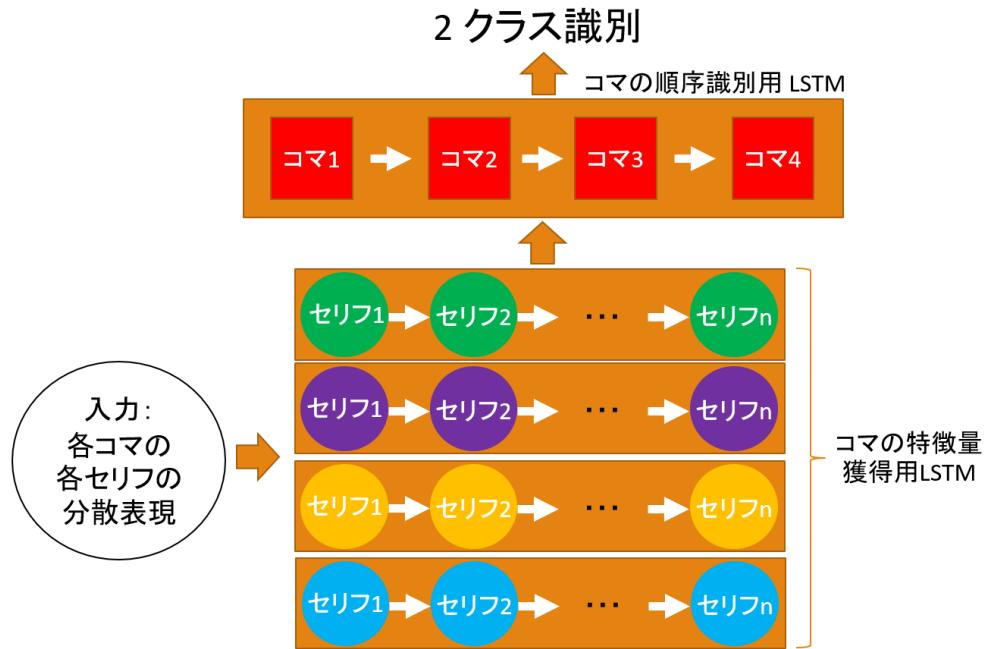


図 3.2: 識別用 LSTM モデル

きない。表 3.2 に実験に使ったデータ中の 1 エピソード内のセリフのあるコマの数をまとめる。これらのデータは今回の自然言語処理手法のみを用いる場合には絶対に正しく識別することができないが、画像やマルチモーダルな識別との比較のために使用した。

また、セリフを有するコマが 2 個以下の場合は解くことができないことは明白である。3 個の場合は不可能ではないが、困難な問題である。すべてのコマにセリフを有する例とセリフを有するコマが 3 個の場合の例を挙げておく。以下に提示する順番はランダムに並び替えられており、実際に計算機が解く問題を体験することができる。

例 1

1. ガキと動物か

男は男で「かわいい」の範囲狭すぎないか

2. コレ？チヨーかわいいでしょ

あゆっぽいでしょ

3. あかわいい
えつ誰？
4. コギャルにとっていいものの形容詞は「かわいい」しかない
んか
コートがかわいいのか

例 2

1. これで若者とシンボクがはかれると思ってんだもんね
ま 1 年に 1 度だしガマンガマン
2. 楽しそうに振るまわなきや!!
オヤジなりに気苦労があるのです
3. とはいえたしだって行きたくない
楽しんでんのはオヤジだけだよね
ほかに旅行行かないし
4. (セリフ無し)

正解は、例 1 は 2 4 3 1、例 2 は 3 1 4 2 である。これらの例が 4 コマ漫画から抜粋されていることを考えれば、起承転結になっているあるいはなっている可能性が高いことがわかり、物語の流れが判然と見えてくる。人間であれば、セリフを持たないコマが 1 つある場合には、物語の流れを理解した上でどこにセリフのないコマを置くと物語が自然になるかを考えることになり、問題がより難しくなっていることがわかる。なお、セリフを持たないコマは、例 2 では中年の上司が部下の文句を聞いてしまったコマである。図 3.3, 3.4 に例 1, 2 の実際の 4 コマ漫画を示す。

セリフを持たないコマが何コマ目に存在しやすい傾向があるのかを調べる。表 3.3 に、セリフを持たないコマが何番目のコマに集中するかをまとめる。「OL ランチ」では、例 2 のような 3 コマ目に存在する場合に加え、1 コマ目にもセリフを持たないコマが集中することがわかる。最も出現回数の多い 3 コマ目でも 45 個中 23 個と約 50 % という偏り具合であり、統計的に問題を解く(つまり文脈を加味せず確率的にもっともらしい)3 コマ目をセリフを持

たないコマとする)場合には、半数近くを間違えることとなる。したがって、計算機も人間と同様に、どこにセリフのないコマを置くと物語が自然になるかを考えることが必須となると考えられる。計算機がこういったデータをうまく識別できるかどうかを確認することも今回の実験のポイントとなる。

表 3.1: モデルのパラメータ

語彙は Doc2Vec , Skip-Thought , BERT で共通のものとするため , 学習済みモデルを使用する BERT に合わせた . ただし , 一部の単語は学習コーパスに存在しておらず , 学習できていない可能性がある .

Doc2Vec	
分散表現次元数	768
ウィンドウサイズ	3
語彙数	32006
Skip-Thought	
分散表現次元数	768
最大入力長	64
語彙数	32006
BERT	
分散表現次元数	768
最大入力長	512
語彙数	32006
Transformer blocks	12
Self attention heads	12
活性化関数	Gaussian error linear units
ドロップアウト	0.1
順序識別用 LSTM	
コマ用 LSTM 次元数	768×2
コマ用 LSTM 方向	双方向
順序識別用 LSTM 次元数	768×2
順序識別用 LSTM 方向	双方向

表 3.2: 実験データにおけるセリフを有するコマ数

セリフを有するコマ数	データの個数
0	1
1	2
2	3
3	45
4	157
計	208

表 3.3: セリフを有するコマが 3 個の各コマ番号におけるセリフのないコマの個数

コマ番号	データの個数
1	15
2	3
3	23
4	4
計	45



図 3.3: 例 1 © さんりょうこ (出典: Manga 109)



図 3.4: 例 2 © さんりようこ (出典 : Manga 109)

表 4.1: 実験データ数

	エピソード数	学習データ数
訓練	158	948
テスト	50	300

4 実験 1

言語情報をもとに文脈から順序を推定するモデルの性能を測定するために，以下の 3 つの実験を試みた．

実験 1.1 連続した 2 コマの順序推定

実験 1.2 4 コマ漫画の順序を不均衡データで学習したモデルで推定

実験 1.3 4 コマ漫画の順序を正例と負例が 1:1 のデータで学習したモデルで推定

ただし，実験 1.3 はモデルの性能を計るという側面に加え，不均衡データである順序推定の学習方法へのアプローチという側面も含まれる．

4.1 実験 1.1

実験 1.1 では，連続した 2 コマに対して順序推定を行った．

4.1.1 実験 1.1 のデータについての注意

表 4.1 にデータ数をまとめる．実験 1.1 では，全 208 エピソードのうち，ランダムで 50 エピソードをテストデータとし，残りを訓練とした．過学習をしていない汎用的なモデルの結果を提示するために 10 分割交差検定を用いた．評価データの結果が最も良いときのモデルが汎用的なモデルであるという仮定のもとで，評価データがもっとよいときの訓練，評価，テスト結果を提示することとなる．実験 1.1 では「OL ランチ」の各エピソードを 1・2 コマ，2・3 コマ，3・4 コマに分割し，それぞれを正順と逆順に並び替える．すなわち，1 エピソードから $3 \times 2 = 6$ 個のデータが作られることとなる．

表 4.2: 実験結果 (BERT)

データ	Accuracy	Precision		Recall		<i>F</i> 値	
		正例	負例	正例	負例	正例	負例
訓練	0.875194	0.883776	0.868232	0.861481	0.888906	0.871970	0.878032
評価	0.632153	0.636382	0.629110	0.620556	0.643750	0.627743	0.635798
テスト	0.547333	0.549135	0.545793	0.530000	0.564667	0.539205	0.554907

4.1.2 実験 1.1 結果 (数値)

表 4.2 , 4.3 , 4.4 に BERT , Skip-Thought , Doc2Vec の分散表現を入力とした場合の実験結果をそれぞれ示す . 正順と逆順の比率は 1:1 なので , ベースラインは 0.500000 である . テストの識別率は最も良い BERT で 0.547333 であった .

実験 1.1 の 2 コマ順序推定は , 4 コマ漫画の順序推定による性能評価の予備実験として考えたが , 問題の難易度としては 4 コマ漫画の順序推定に比べ劣るものではなかった . 例えば , 3.4 節で提示した例 1 と例 2 を考える . 例 1 においては , 1 · 2 · 3 · 4 のすべてのコマで順序推定を行った場合は一意に順序が定まるが , 2 · 3 コマの 2 つのコマで順序を推定する場合 , 人間でも順序を推定することが困難である . 例 2 の 4 コマ漫画においても , 今回は言語情報だけなのでセリフのない 3 コマ目が何を表現しているのかわからず , セリフのない 3 コマ目を含む , 2 · 3 コマと 3 · 4 コマの推定は非常に困難なものである .

この実験 1.1 の結果から , 実験 1.2 や 実験 1.3 の 4 コマ漫画の順序推定に対する直接的な知見というものが得られるものではなかった . しかしながら , 事前情報として持つコマ数が少なすぎる (文脈情報が少なすぎる) と順序推定が難しくなるということを示唆した結果であり , 4 コマ漫画以外の漫画に対し順序推定を行う場合に , 何コマを事前情報として加味するかということが重要になってくるということがわかる .

表 4.3: 実験結果 (Skip-Thought)

データ	Accuracy	Precision		Recall		<i>F</i> 値	
		正例	負例	正例	負例	正例	負例
訓練	0.843307	0.843483	0.843295	0.843414	0.843199	0.843399	0.843196
評価	0.622986	0.625731	0.620745	0.613611	0.632361	0.619354	0.626264
テスト	0.507333	0.507264	0.507398	0.502000	0.512667	0.504515	0.509917

表 4.4: 実験結果 (Doc2Vec)

データ	Accuracy	Precision		Recall		<i>F</i> 値	
		正例	負例	正例	負例	正例	負例
訓練	0.879867	0.879264	0.881046	0.879985	0.879749	0.879453	0.880225
評価	0.611042	0.611346	0.610951	0.610139	0.611944	0.610553	0.611260
テスト	0.543667	0.543459	0.543945	0.542667	0.544667	0.542946	0.544189

表 4.5: 実験データ数

	エピソード数	学習データ数
訓練	158	3792
テスト	50	1200

表 4.6: 実験結果 (BERT)

データ	Accuracy	Precision		Recall		<i>F</i> 値	
		正例	負例	正例	負例	正例	負例
訓練	0.927418	0.429771	0.997721	0.951468	0.926372	0.572280	0.959903
評価	0.885417	0.184639	0.970034	0.348333	0.908768	0.209715	0.937110
テスト	0.873333	0.069858	0.960909	0.152000	0.904696	0.087426	0.930709

4.2 実験 1.2

実験 1.2 では不均衡データを用いて学習したモデルで，4 コマ漫画の正しい順序を選択できるかどうかを確認する。

4.2.1 実験 1.2 のデータについての注意

表 4.5 にデータ数をまとめる。実験 1.2 では，全 208 エピソードのうち，ランダムで 50 エピソードをテストデータとし，残りを訓練とした。過学習をしていない汎用的なモデルの結果を提示するために 10 分割交差検定を用いた。実験 1.2 では，評価データにおいて，Accuracy ではなく，*F* 値が最も高いエポックの結果を示す。実験 1.2 では「OL ランチ」の各エピソードのコマを(正例 1 つを含む)24 通りに並び替え，入力された順番が正例か負例かを識別するので，実際に学習に使用した訓練データは $158 \times 24 = 3792$ 個(いくつかは評価データとして用いる)でテストが $50 \times 24 = 1200$ 個であること，正例と負例が 1 : 23 の比率で存在する不均衡データであることに注意する。

表 4.7: 実験結果 (Skip-Thought)

データ	Accuracy	Precision		Recall		<i>F</i> 値	
		正例	負例	正例	負例	正例	負例
訓練	0.935176	0.420512	0.994871	0.889596	0.937158	0.559894	0.964885
評価	0.900330	0.199050	0.970742	0.355833	0.924004	0.234739	0.946182
テスト	0.885667	0.058635	0.960063	0.120000	0.918957	0.075668	0.938605

表 4.8: 実験結果 (Doc2Vec)

データ	Accuracy	Precision		Recall		<i>F</i> 値	
		正例	負例	正例	負例	正例	負例
訓練	0.955560	0.565794	0.998676	0.971007	0.954889	0.692330	0.975923
評価	0.880538	0.171670	0.967679	0.305833	0.905525	0.203951	0.932135
テスト	0.872917	0.054475	0.959837	0.126000	0.905391	0.069553	0.929445

4.2.2 実験 1.2 結果 (数値)

表 4.6 , 4.7 , 4.8 に BERT , Skip-Thought , Doc2Vec の分散表現を入力とした場合の実験結果をそれぞれ示す . 正例と負例の比率は 1 : 23 なので , 今回の実験の Accuracy のベースライン (負例のみを選択) は $23/24 = 0.958333$ である .

テストデータの *F* 値に着目すると BERT , Skip-Thought , Doc2Vec の順番で結果が良かった . 最も良い BERT でも正例の *F* 値が 0.087426 であり , 決して高いと言えない結果となつたが , データが非常に不均衡であることに加え , 日本語コミックのコマの順序識別に対して , 自然言語処理の方面からアプローチした研究はなく , 実験 1.1 と同様に , 比較対象が存在しないためこの結果の判断は難しい .

なお , 表 4.6 , 4.7 , 4.8 のテストデータの正例 *F* 値が低い値であるが , 学習していないわけではない . ここで表 4.9 に BERT , Skip-Thought , Doc2Vec の正例選択数および選択率をそれぞれ提示しておく . 正例の割合が $1/24 \sim 0.041667$

表 4.9: テストデータの正例選択率および選択数

モデル	選択率	選択数	全データ数
BERT	0.07992	95.9	1200
Skip-Thought	0.1010	121.2	1200
Doc2Vec	0.1226	147.1	1200
ベースライン	0.0417	50.0	1200

であることを考えれば、表 4.9 の値は低い値ではない。

4.2.3 実験 1.2 結果 (4 コマ正解数)

次に 4 コマ漫画の正解数について見ていく。ここまで今回の実験の Accuracy や F 値等について確認してきたが、今回のコマの順序識別において、真に見るべきは正しい順序や間違った順序に並び替えられた 4 コマ漫画の中から正解を選択できるかどうかである。

モデルの識別層から正例である確率を獲得し、これらを比較した。最も高い確率で正例だと判断した順序をモデルの答えとする。

なお、4 コマ漫画の順序を **1324** といった 4 枠の数字で表すことにする。この場合、4 コマ漫画の 1 コマ目、3 コマ目、2 コマ目、4 コマ目の順で並んでいる。また、**1** や **4** のように文字が赤くなっている数字が正解の順序と合致しているコマであり、黒字で書かれている **2** や **3** が正解の順序と異なっているコマである。

一対比較 まずは正例と負例 (23 通り) のそれぞれを比較した。表 4.12 に BERT, Skip-Thought, Doc2Vec の一対比較時の正しい順序としての選択率を記す。表 4.12 より、BERT の分散表現を用いた場合、最も選択率が高かったのは **2431** で 69.80 %、最も選択率が悪かったのは **3214** で 51.20 % であった。Skip-Thought の分散表現を用いた場合、最も選択率が高かったのは **3142** で 63.80 %、最も選択率が悪かったのは **3124** で 48.20 % であった。Doc2Vec の分散表現を用いた場合、最も選択率が高かったのは **3421** で 69.80 %、最も選択率が悪かったのは **2314** で 47.40% であった。平均では、最も選択率が高

かったのは 2431 で 65.33 % , 最も選択率が悪かったのは 2314 で 51.00 % であった .

実際に解けたデータを確認すると , 学習データやエポックごとに解けたものに違いがあることが確認できるが , 10 分割交差検定をしているため , 結果のぶれはある程度抑えられた結果である . モデルごとのアルゴリズムの違いからか , 選択率が高かった順序や低かった順序はモデルごとに異なっていたが , BERT と Skip-Thought と Doc2Vec を比べると , 平均選択率は BERT が最も高い . このため , 分散表現を用いて 4 コマ漫画の順序識別をする場合には BERT が最もよいと考えられる . ただし , 本実験設定では文やコマ間の論理的な関係や起承転結のような 4 コマ漫画の構造を理解しているかどうかを考察することが難しい . これらを調査するためには , 実験設定を変える必要がある . 例えば , 本研究でデータとして用いた OL ランチはどのエピソードも起承転結からなっていたが , 4 コマ漫画によっては起承転結ではなく , 出オチであったり , そもそもオチが存在しないものもある . 複数の構造の 4 コマ漫画を組み合わせることで , 起承転結といった論理構造への考察は可能であると考えられる . このような調査は今後の課題とする .

次に , 考察のしやすいセリフの数や長さといった点から考察していく . 表 4.10 に「OL ランチ」の各コマの長さとセリフの数の平均と分散をまとめる .

表 4.10 より ,

考察 1 セリフの数の点では , 1・3 コマと 2・4 コマが類似している

考察 2 セリフの長さという点では , 4 コマ目が 1・2・3 コマとかなり異なっている

考察 3 4 コマ目が最も単語数が多く , セリフの数も多いため判断しやすいことから 4 コマ目が真の順序と不一致な場合 , 4 コマ目によって判断できる可能性が高い

考察 4 3 コマ目が真の順序と一致している場合 , 他コマの判断がしやすく識別しやすい

と考えられる .

表 4.10: 「OL ランチ」の各コマの長さとセリフの数の平均と分散

セリフの単語数		
コマ	平均	分散
全	7.57635	16.9825
1	7.35119	13.4838
2	7.55216	13.4534
3	7.51497	16.6510
4	7.88710	20.0771
セリフの数		
コマ	平均	分散
全	1.79928	0.809471
1	1.61538	0.736686
2	1.88942	0.781042
3	1.60577	0.806121
4	2.08654	0.752126

考察 1 における，1・3 コマの類似は，BERT の分散表現を用いた場合における 3214 からも見て取れる．1 コマ目と 3 コマ目がセリフの数という点で類似しているのは，セリフのないコマの影響であると考えられる．前述の通り，セリフのないコマは 1 コマ目と 3 コマ目に集中している．

考察 2 においてセリフの長さについて述べたが，セリフの長さという情報に関しては，本提案手法の入力は分散表現であるため，直接的な情報として保持していない．ただし，分散表現がセリフの長さという情報を保持している可能性は高く，例えば，文の長さが同じ文は類似している傾向がある．図 4.1 に，BERT による分散表現をプロットしたものを示す．BERT の分散表現は本来 768 次元であるが，t-SNE (t-distributed Stochastic Neighbor Embedding)^[27] によって 2 次元に次元圧縮した．図 4.1 より，同じ色のプロットが固まっていることから，分の長さによる影響が現れていることがわかる．

考察 3 と考察 4 における 3 コマ目と 4 コマ目の真の順序との一致・不一致による影響を調べる．表 4.11 に，一対比較時の 3 コマ目が真の順序と一致な

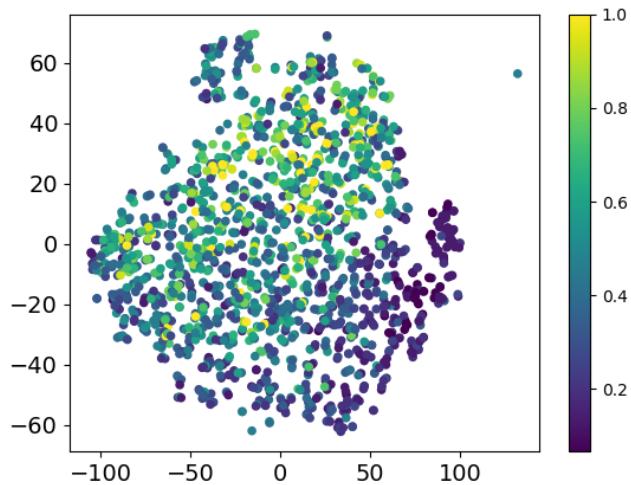


図 4.1: BERT の分散表現
縦軸と横軸は 2 次元に次元圧縮されたときの値である

表 4.11: 3 コマ目一致および不一致と 4 コマ目一致および不一致の平均

	BERT	Skip-Thought	Doc2Vec
3 コマ目一致	0.5901	0.5820	0.5605
3 コマ目不一致	0.5684	0.5652	0.5760
4 コマ目一致	0.5825	0.5673	0.5631
4 コマ目不一致	0.5944	0.6152	0.5672
全体の平均	0.5881	0.5790	0.5657

ものの平均と不一致ものの平均，4 コマ目が真の順序と一致ものの平均と不一致ものの平均についてまとめる．表 4.11 より，考察 3 と考察 4 で述べた通り，3 コマ目が一致しているものと 4 コマ目が不一致ものは全体の平均より正解率が高いことがわかる．

これらの考察から，提案手法が物語のストーリーをどの程度理解して答えを導き出しているのかはわからないが，各コマのセリフの数や長さ（アルゴリズムによってはセリフの長さが影響の大きい分散表現になっている場合が見られる）といった点から統計的に解きやすい順序と解きにくい順序が存在するのではないかと考えられる．ただし，今回の実験での解きやすい順序が 4

コマ漫画あるいはそれ以外の漫画においても解きやすいとは言えない。例えば、今回のデータとして用いた「OL ランチ」は、起承転結がはっきりとした漫画であると言えるが、すべての 4 コマ漫画が起承転結が明確な作品ではなく、物語に起伏やオチがない場合も見られる。また、セリフのないコマを多用するなどの特徴のある作品も見られるため、複数漫画で学習する場合には、4 コマ漫画の普遍的な特徴を学ぶことに加え、推定したい 4 コマ漫画に特化した特徴も捉えられるようなモデルを構築していく必要があると考えられる。

完全不一致比較 順序が 2143 や 2341 のような完全不一致と正しい順序 (1234) の 10 通りの順序を比較した。表 4.13 に BERT, Skip-Thought, Doc2Vec の不完全一致比較時の選択率を記す。ベースラインが 0.1000 なので、どのモデルでもベースラインを越す識別率が出せた。

24 通り 4 コマ漫画の順序として、正順を含む 24 通りのすべての順序で比較した。表 4.14 に BERT, Skip-Thought, Doc2Vec の全通りの選択率を記す。一対比較時に選択率が低かった 3124 や 3214 は、全通りで比較した場合も低いことがわかる。完全不一致と同様に、どのモデルでもベースラインを越す識別率が出せた。

表 4.12: 4 コマ順序推定選択率 (一対比較)

この表では、正しい順序の 4 コマ漫画の正例の確率とランダムに並び替えられた 4 コマ漫画の正例の確率を一対比較し、値の大きいものを正しい順序として選択した場合の各順序の割合を示す。つまり、この表からは、例えば、BERT の分散表現を用いて、1234 と 1243 を比べた場合には、モデルは 54.80% の 4 コマ漫画の順序を正しく推定できたのだということがわかる。なお、これはテストデータの結果であり、ベースラインは 50 %で、平均は 3 つのモデルの識別率の平均を示している。

順序	BERT	Skip-Thought	Doc2Vec	平均	順序	BERT	Skip-Thought	Doc2Vec	平均
1234	0.5480	0.5880	0.5420	0.5593	1234	0.5680	0.6380	0.5400	0.5820
1243	0.4520	0.4120	0.4580	0.4407	3142	0.4320	0.3620	0.4600	0.4180
1234	0.5440	0.5000	0.5080	0.5173	1234	0.5120	0.5220	0.5100	0.5147
1324	0.4560	0.5000	0.4920	0.4827	3214	0.4880	0.4780	0.4900	0.4853
1234	0.5440	0.6120	0.5620	0.5727	1234	0.6700	0.6120	0.5900	0.6240
1342	0.4560	0.3880	0.4380	0.4273	3241	0.3300	0.3880	0.4100	0.3760
1234	0.5620	0.5460	0.5840	0.5640	1234	0.5540	0.6000	0.6040	0.5860
1423	0.4380	0.4540	0.4160	0.4360	3412	0.4460	0.4000	0.3960	0.4140
1234	0.5560	0.6300	0.5880	0.5913	1234	0.6800	0.5780	0.6400	0.6327
1432	0.4440	0.3700	0.4120	0.4087	3421	0.3200	0.4220	0.3600	0.3673
1234	0.5760	0.5420	0.4960	0.5380	1234	0.5860	0.5480	0.5740	0.5693
2134	0.4240	0.4580	0.5040	0.4620	4123	0.4140	0.4520	0.4260	0.4307
1234	0.6020	0.5860	0.5380	0.5753	1234	0.5680	0.6320	0.5740	0.5913
2143	0.3980	0.4140	0.4620	0.4247	4132	0.4320	0.3680	0.4260	0.4087
1234	0.5420	0.5140	0.4740	0.5100	1234	0.5600	0.5460	0.5560	0.5540
2314	0.4580	0.4860	0.5260	0.4900	4213	0.4400	0.4540	0.4440	0.4460
1234	0.6620	0.6460	0.5700	0.6260	1234	0.6500	0.6120	0.6200	0.6273
2341	0.3380	0.3540	0.4300	0.3740	4231	0.3500	0.3880	0.3800	0.3727
1234	0.5780	0.5560	0.5780	0.5707	1234	0.5360	0.5960	0.5900	0.5740
2413	0.4220	0.4440	0.4220	0.4293	4312	0.4640	0.4040	0.4100	0.4260
1234	0.6980	0.6340	0.6280	0.6533	1234	0.6520	0.5980	0.6040	0.6180
2431	0.3020	0.3660	0.3720	0.3467	4321	0.3480	0.4020	0.3960	0.3820
1234	0.5780	0.4820	0.5420	0.5340	各モデルの平均選択率				
3124	0.4220	0.5180	0.4580	0.4660	1234	0.5881	0.5790	0.5657	

表 4.13: 4 コマ順序推定選択率(不完全一致比較)

ベースラインは $1/10 = 0.1000$ である。

順序	BERT	Skip-Thought	Doc2Vec	合計
1234	0.2540	0.2260	0.2280	0.2360
2143	0.0720	0.0480	0.0920	0.0707
2341	0.0460	0.0600	0.1100	0.0720
2413	0.1220	0.1260	0.0900	0.1127
3142	0.0960	0.0820	0.0900	0.0893
3412	0.0880	0.1000	0.0560	0.0813
3421	0.0880	0.1420	0.0600	0.0967
4123	0.0480	0.1060	0.1120	0.0887
4312	0.1400	0.0580	0.1040	0.1007
4321	0.0460	0.0520	0.0580	0.0520

表 4.14: 4 コマ順序推定選択率 (全通り比較)

ベースラインは $1/24 = 0.0417$ である。

順序	BERT	Skip-Thought	Doc2Vec	合計
1234	0.0920	0.0860	0.0900	0.0893
1243	0.0580	0.0500	0.0420	0.0500
1324	0.0480	0.0540	0.0540	0.0520
1342	0.0340	0.0580	0.0540	0.0487
1423	0.0380	0.0500	0.0320	0.0400
1432	0.0420	0.0320	0.0360	0.0367
2134	0.0460	0.0460	0.0740	0.0553
2143	0.0360	0.0240	0.0320	0.0307
2314	0.0660	0.0300	0.0620	0.0527
2341	0.0120	0.0160	0.0200	0.0160
2413	0.0280	0.0380	0.0220	0.0293
2431	0.0500	0.0500	0.0340	0.0447
3124	0.0360	0.0740	0.0460	0.0520
3142	0.0380	0.0260	0.0380	0.0340
3214	0.0660	0.0480	0.0460	0.0533
3241	0.0400	0.0260	0.0260	0.0307
3412	0.0480	0.0320	0.0380	0.0393
3421	0.0240	0.0280	0.0240	0.0253
4123	0.0160	0.0340	0.0440	0.0313
4132	0.0300	0.0280	0.0500	0.0360
4213	0.0360	0.0720	0.0480	0.0520
4231	0.0400	0.0420	0.0420	0.0413
4312	0.0440	0.0240	0.0300	0.0327
4321	0.0320	0.0320	0.0160	0.0267

表 4.15: 実験データ数

	エピソード数	学習データ数
訓練	158	316
テスト	50	1200

4.3 実験 1.3

実験 1.3 では学習データを不均衡ではなく、1:1 として学習したモデルで、4 コマ漫画の正しい順序を選択できるかどうかを確認した。実験 1.2 では不均衡なデータを用いて学習したが、その場合、不均衡データでモデルをどのように学習していくかが重要となってくる。実験 1.3 では学習するデータ(順序パターン)が少なくなるが不均衡データではないため学習が容易であると考えられ、データがどうしても不均衡となる順序推定の学習に対する知見を得るのが目的である。

4.3.1 実験 1.3 のデータについての注意

表 4.15 にデータ数をまとめた。実験 1.3 では、全 208 エピソードのうち、ランダムで 50 エピソードをテストデータとし、残りを訓練とした。実験 1, 2 と同様に、過学習をしていない汎用的なモデルの結果を提示するために 10 分割交差検定を用いた。

実験 1.3 では「OL ランチ」の各エピソードから正例(1234)と負例を 1 つずつ用意した。負例は、正例を含まない 23 通りのうちからランダムに 1 つ選んだ順序であった(エピソードごとに 23 通りから 1 つをランダムに選ぶので、実験 1.2 と同様に、モデルは 24 通りすべてを学習した)。ただし、学習に影響しない評価データとテストデータは 24 通りのすべての順序を用意した。つまり、評価とテストは 1 : 23 の比率で存在する不均衡データである。実際に学習に使用した訓練データは $158 \times 2 = 316$ 個(いくつかは評価データとして用い、評価は 24 通りのデータで行う)、テストが $50 \times 24 = 1200$ 個であった。

表 4.16: 実験結果 (BERT)

データ	Accuracy	Precision		Recall		<i>F</i> 値	
		正例	負例	正例	負例	正例	負例
訓練	0.945117	0.953465	0.940732	0.935256	0.954979	0.943047	0.946836
評価	0.633715	0.068309	0.973908	0.607917	0.634837	0.122477	0.766728
テスト	0.603500	0.050268	0.964133	0.476000	0.609043	0.090676	0.744235

表 4.17: 実験結果 (Skip-Thought)

データ	Accuracy	Precision		Recall		<i>F</i> 値	
		正例	負例	正例	負例	正例	負例
訓練	0.943058	0.952344	0.937043	0.932829	0.953286	0.941532	0.944342
評価	0.598872	0.065415	0.973639	0.629583	0.597536	0.118066	0.737202
テスト	0.580833	0.050564	0.964296	0.504000	0.584174	0.091598	0.724530

4.3.2 実験 1.3 結果 (数値)

表 4.16 , 4.17 , 4.18 に BERT , Skip-Thought , Doc2Vec の分散表現を入力とした場合の実験結果をそれぞれ示す . 評価データとテストデータにおいては , 正例と負例の比率は 1 : 23 である . 実験 1.2 に比べ , 評価やテストの Accuracy が低く , 正例の Recall が高いことが確認できる . 表 4.19 にテストデータの正例選択率および選択数をまとめる . 表 4.19 より , どのモデルも選択率が 3 割を超えており , 実験 1.2 よりも正例と選択しやすいとわかる . 実験 1.2 の結果に比べ , この結果は Recall が高いことで *F* 値も高い一方で , Accuracy が低く , 一概に実験 1.2 よりも良いとは言えるかどうか難しい .

4.3.3 実験 1.3 結果 (4 コマ正解数)

次に 4 コマ漫画の正解数について見ていく .

モデルの識別層から正例である確率を獲得し , これらを比較した . 最も高い確率で正例だと判断した順序をモデルの答えとした .

表 4.18: 実験結果 (Doc2Vec)

データ	Accuracy	Precision		Recall		F 値	
		正例	負例	正例	負例	正例	負例
訓練	0.966052	0.964417	0.968298	0.96818	0.963925	0.966141	0.965951
評価	0.644097	0.068114	0.972498	0.57875	0.646938	0.121401	0.774957
テスト	0.607750	0.050251	0.963984	0.47000	0.613739	0.090561	0.748037

表 4.19: テストデータの正例選択率および選択数

モデル	選択率	選択数	全データ数
BERT	0.3393	407.1	1200
Skip-Thought	0.4008	481.0	1200
Doc2Vec	0.4042	485.0	1200
ベースライン	0.0417	50.0	1200

一対比較 まずは正例と負例 (23 通り) のそれぞれを比較した。表 4.20 に BERT , Skip-Thought , Doc2Vec の一対比較時の選択数をそれぞれ記す。各モデルの正例の選択率を確認すると、すべてのモデルで実験 1.2 よりも値が低く、実際の順序推定では不均衡データで学習した場合のほうが良かった。実験 1.2 では不均衡データをどのように学習するのかを考えなければならない一方、データ数が多い事に加え、負例のパターンを多く学習できる実験 1.2 の方がよい結果となった。ただし、学習データの数は実験 1.2 に比べ、1/12 倍であり、学習コストはこちらのほうが低かった。

完全不一致比較 順序が 2143 や 2341 のような完全不一致と正しい順序 (1234) の 10 通りの順序を比較した。表 4.21 に BERT , Skip-Thought , Doc2Vec の不完全一致比較時の選択率を記す。

こちらでもベースラインを超える結果となった。不完全一致の場合は、実験 1.2 よりも性能が良いため、正しい順序と大きく異なるような順序の場合には、実験 1.3 のような条件での学習が良い可能性がある。

表 4.20: 4 コマ順序推定選択率 (一対比較)

この表では、正しい順序の 4 コマ漫画の正例の確率とランダムに並び替えられた 4 コマ漫画の正例の確率を一対比較し、値の大きいものを正しい順序として選択した場合の各順序の割合を示す。つまり、この表からは、例えば、BERT の分散表現を用いて、1234 と 1243 を比べた場合には、モデルは 57.00% の 4 コマ漫画の順序を正しく推定できたのだということがわかる。なお、これはテストデータの結果であり、ベースラインは 50 %で、平均は 3 つのモデルの識別率の平均を示している。

順序	BERT	Skip-Thought	Doc2Vec	平均	順序	BERT	Skip-Thought	Doc2Vec	平均
1234	0.5700	0.5980	0.5380	0.5687	1234	0.5560	0.5820	0.5320	0.5567
1243	0.4300	0.4020	0.4620	0.4313	3142	0.4440	0.4180	0.4680	0.4433
1234	0.5600	0.5000	0.4960	0.5187	1234	0.4940	0.4640	0.5620	0.5067
1324	0.4400	0.5000	0.5040	0.4813	3214	0.5060	0.5360	0.4380	0.4933
1234	0.5740	0.5840	0.5440	0.5673	1234	0.6200	0.6080	0.6320	0.6200
1342	0.4260	0.4160	0.4560	0.4327	3241	0.3800	0.3920	0.3680	0.3800
1234	0.5780	0.5480	0.5300	0.5520	1234	0.5600	0.5600	0.5480	0.5560
1423	0.4220	0.4520	0.4700	0.4480	3412	0.4400	0.4400	0.4520	0.4440
1234	0.5540	0.5640	0.5300	0.5493	1234	0.6360	0.5940	0.6180	0.6160
1432	0.4460	0.4360	0.4700	0.4507	3421	0.3640	0.4060	0.3820	0.3840
1234	0.5360	0.5220	0.5060	0.5213	1234	0.5860	0.5620	0.5240	0.5573
2134	0.4640	0.4780	0.4940	0.4787	4123	0.4140	0.4380	0.4760	0.4427
1234	0.5800	0.5840	0.5420	0.5687	1234	0.5620	0.5880	0.5340	0.5613
2143	0.4200	0.4160	0.4580	0.4313	4132	0.4380	0.4120	0.4660	0.4387
1234	0.5000	0.4620	0.5360	0.4993	1234	0.5760	0.5760	0.5400	0.5640
2314	0.5000	0.5380	0.4640	0.5007	4213	0.4240	0.4240	0.4600	0.4360
1234	0.6340	0.6240	0.6400	0.6327	1234	0.6460	0.5960	0.6380	0.6267
2341	0.3660	0.3760	0.3600	0.3673	4231	0.3540	0.4040	0.3620	0.3733
1234	0.5680	0.5460	0.5400	0.5513	1234	0.5580	0.5800	0.5620	0.5667
2413	0.4320	0.4540	0.4600	0.4487	4312	0.4420	0.4200	0.4380	0.4333
1234	0.6360	0.5960	0.6400	0.6240	1234	0.6500	0.6100	0.6280	0.6293
2431	0.3640	0.4040	0.3600	0.3760	4321	0.3500	0.3900	0.3720	0.3707
1234	0.5280	0.5220	0.4840	0.5113	各モデルの正解選択率				
3124	0.4720	0.4780	0.5160	0.4887	1234	0.5766	0.5639	0.5584	

表 4.21: 4 コマ順序推定選択率 (不完全一致比較)

ベースラインは $1/10 = 0.1000$ である .

順序	BERT	Skip-Thought	Doc2Vec	合計
1234	0.2980	0.2920	0.3080	0.2993
2143	0.0740	0.0640	0.0740	0.0707
2341	0.0860	0.0620	0.0640	0.0707
2413	0.0940	0.1320	0.0820	0.1027
3142	0.1020	0.0880	0.1300	0.1067
3412	0.0700	0.0700	0.0620	0.0673
3421	0.0540	0.1000	0.0680	0.0740
4123	0.0740	0.0820	0.1060	0.0873
4312	0.1040	0.0620	0.0780	0.0813
4321	0.0440	0.0480	0.0280	0.0400

24 通り 4 コマ漫画の順序として , 正順を含む 24 通りのすべての順序で比較した . 表 4.22 に BERT , Skip-Thought , Doc2Vec の全通りの選択率を記す . 一対比較時に選択率が低かった 3124 や 3214 は , 全通りで比較した場合も低いことがわかる .

こちらは , 実験 1.2 よりも正しい順序の選択率が低かったが , ベースラインは超えることができた .

表 4.22: 4 コマ順序推定選択率 (全通り比較)

ベースラインは $1/24 = 0.0417$ である。

順序	BERT	Skip-Thought	Doc2Vec	合計
1234	0.0660	0.0660	0.0760	0.0693
1243	0.0360	0.0420	0.0540	0.0440
1324	0.0380	0.0540	0.0640	0.0520
1342	0.0460	0.0240	0.0540	0.0413
1423	0.0240	0.0300	0.0420	0.0320
1432	0.0280	0.0360	0.0420	0.0353
2134	0.0480	0.0560	0.0500	0.0513
2143	0.0380	0.0360	0.0280	0.0340
2314	0.0660	0.0740	0.0620	0.0673
2341	0.0320	0.0280	0.0240	0.0280
2413	0.0480	0.0640	0.0280	0.0467
2431	0.0320	0.0600	0.0220	0.0380
3124	0.0480	0.0520	0.0460	0.0487
3142	0.0340	0.0340	0.0480	0.0387
3214	0.0760	0.0420	0.0440	0.0540
3241	0.0280	0.0220	0.0220	0.0240
3412	0.0460	0.0400	0.0320	0.0393
3421	0.0260	0.0340	0.0240	0.0280
4123	0.0420	0.0320	0.0580	0.0440
4132	0.0500	0.0400	0.0580	0.0493
4213	0.0340	0.0480	0.0420	0.0413
4231	0.0440	0.0320	0.0320	0.0360
4312	0.0540	0.0260	0.0300	0.0367
4321	0.0160	0.0280	0.0180	0.0207

5 実験 2

ここまで人工知能を用い漫画の順序を推定してきたが、ベースラインを超える識別率を得ることができた。そこで、追加実験として、漫画に登場するキャラクターの各セリフの感情識別を追加実験として行う。

5.1 実験 2 で用いるデータセット

4コマ漫画を対象としたデータセットはいくつか存在するが、感情識別では上野によって作られた4コマストーリーデータセット^[21]を用いた。このデータセットはコミック工学発展のために研究者が漫画の執筆から関わった世界初の研究用のデータセットであり、いくつかの特徴がある。

Manga 109といった市販コミックによって構成されたデータセットとは異なり、4コマストーリーデータセットのデータは本データセットのために幾人かの漫画家によって描き下ろされている。市販されたコミックをデータとした場合、著作権などの問題に加え、計算機上で扱うためのメタデータが少なく、コミックの意味理解を目的とした研究には適さないという問題がある。例えばコミックに登場するキャラクターの感情は明示されていないため、読者によるアノテートによってラベルを付与する必要があるが、アノテートされたラベルが漫画家の意図とは異なる可能性を否定できない。また、マルチモーダルでストーリーの解析をする際にオリジナリティの観点から同一プロットを複数の漫画家が描くことは稀有なため、そういうデータの収集に基づく研究は困難である。4コマストーリーデータセットはそういう問題点を解決するために作られたデータセットである。また、4コマ漫画のあらすじや状況説明文といったメタ情報も含まれており、実験 2 では、セリフのみならず状況説明文も事前情報として用いた。

上野は4コマ漫画の構造を、

- 一般：標準的な起承転結をもつ
- 繰り返し：1, 2コマ間の類似が3, 4コマ間でも起きる
- 出オチ：1コマ目におかしな絵が描かれてオチがある



図 5.1: 4 コマ漫画ストーリーデータセットのデータの一例

- (c) 作画: 鈴木市規 (シナリオ: (株) スポマ 播村早紀/豊橋技術科学大学 上野未貴)
(c) 作画: 浦田カズヒロ (シナリオ: (株) スポマ 播村 早紀/豊橋技術科学大学 上野 未貴)

- タイトルオチ : 最後にタイトルを見返してオチがわかる
- 再帰 : 4 コマ目から 1 コマ目に戻り話として成立する
- 参照 : 1 つ以上前の話の続きを話となる
- 連続 : 連続した 4 コマを 2 話並べて 8 コマで話となる

と定義し、これに従ってデータセットを作成している。現在は、同一のストーリーを 4 コマ目がオチとなる一般と出オチの 2 つの構造から描いたものがデータとして存在しており、本研究では計算機でこれら 2 つの構造の把握することを目的としている。

表 5.1: 実験 2 のデータ数

	データ数
ニュートラル	51
その他	101
計	152

5.2 データについて

4コマ漫画ストーリーデータセットでは、各セリフには、ニュートラル・喜楽・恐怖・嫌悪・悲哀・憤怒・驚愕の7種類の感情ラベルが付与されている。実験2では、データ数の問題から、「ニュートラル」とそれ以外のラベルの集合である「その他」2種類のラベルに振り直して、実験した。実験2では、10分割交差検定によって、学習データを訓練データとテストデータに分割した。

5.3 モデル

実験2では、登場キャラクターのセリフの感情を2種類のモデルを使い推定する。1つ目は、推定対象のセリフのみを入力とし推定するモデルで、2つ目は、推定対象のセリフの直前のセリフや状況説明文を事前情報として用いるモデルである。2つ目の事前情報を用いたモデルでは、どれくらいの事前情報を用いるかという任意性がある。実験2では、セリフあるいは状況説明文を1つ用いた場合とセリフあるいは状況説明文を合わせて2つ用いた場合、セリフあるいは状況説明文を合わせて3つ用いた場合の3パターンで実験する(以下、それぞれ、1事前情報モデル、2事前情報モデル、3事前情報モデル、これらの総称として事前情報ありモデルと呼び、1つの事前情報を用いないモデルを事前情報なしモデルと呼ぶ)。なお、Doc2Vecは日本語の Wikipedia⁵ や 青空文庫⁶を用いて学習した。図5.2、5.3に、それぞれ、事前情報なしモデルの概要と事前情報ありモデルの概要を示す。

表5.2に、実験2で用いたモデルのパラメータを示す。

⁵<https://ja.wikipedia.org/>

⁶<https://www.aozora.gr.jp/>

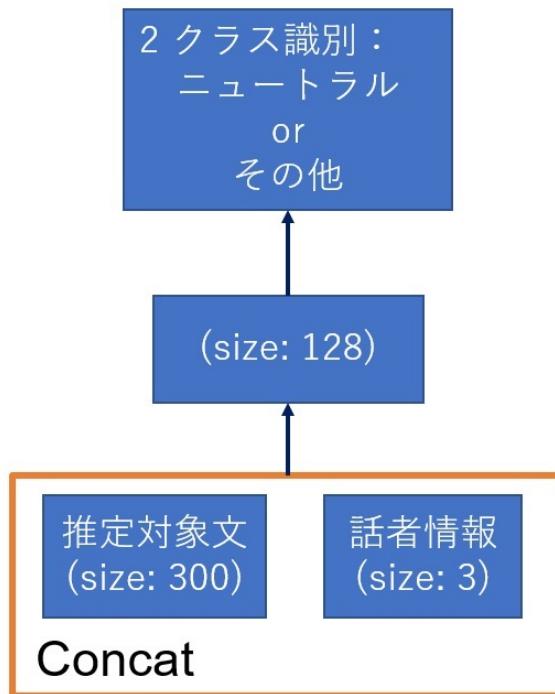


図 5.2: 事前情報なしモデル

表 5.2: モデルのパラメータ

Doc2Vec	
分散表現次元数	300
ウィンドウサイズ	3
語彙数	42375

5.4 実験 2 の結果と考察

表 5.3 に、実験結果を示す。事前情報なしモデル以外の 3 つのモデルは、訓練データの Accuracy が 0.9 ほどであるが、これは過学習を防ぐために用いたドロップアウトによる影響である。最もテストデータでの結果がよいのが、1 事前情報モデルであり、0.767 の識別率を出せた。しかしながら、データの偏りからどのモデルでも「その他」の Precision が高かった中、最もニュートラルの Precision が高かったのが 2 事前情報モデルであり、感情識別ではセリフや状況説明文を 2 つ用いた場合が最も文脈情報を加味できていると考えられ

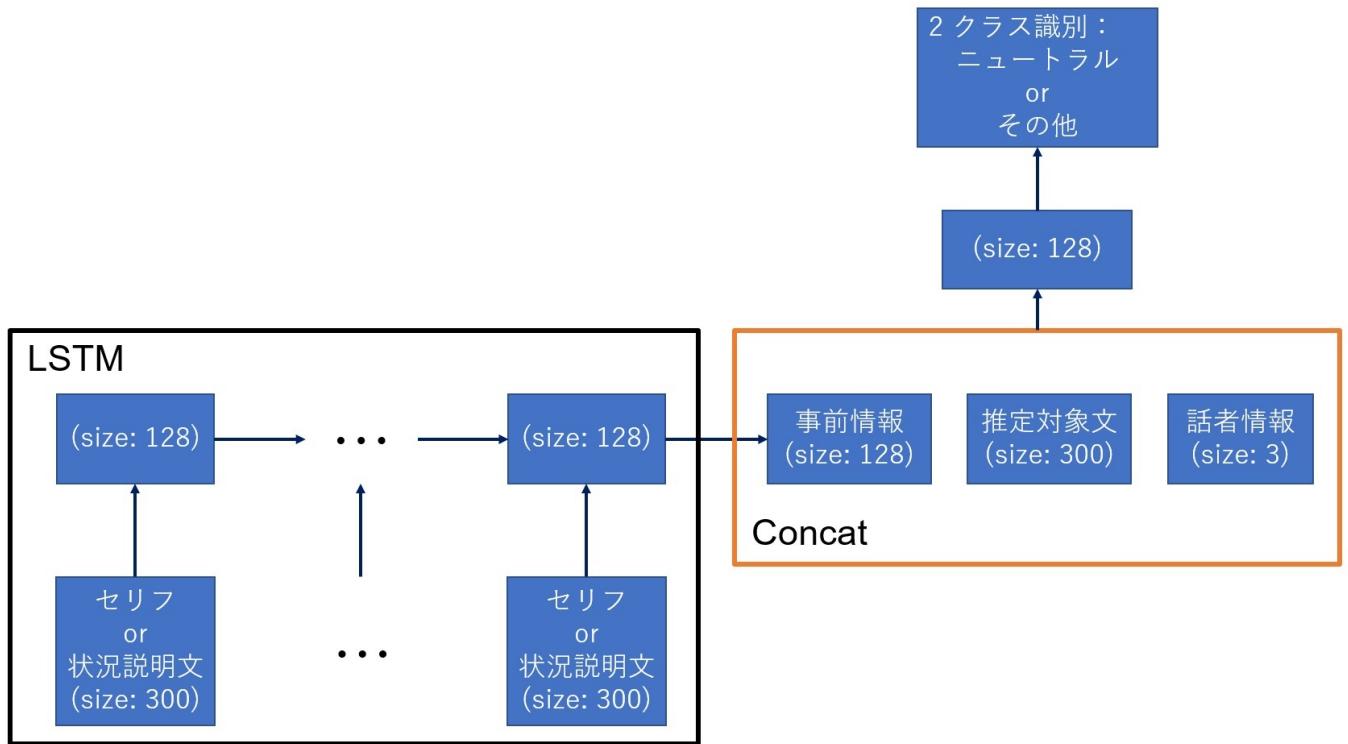


図 5.3: 事前情報ありモデル

る。また、3 事前情報モデルの実験結果からわかるように、情報の多さが必ずしも実験結果に良い影響を及ぼすわけではなく、ノイズとなるかもしれない可能性を示唆している。これは、実験 1.1 からもわかる通り、順序を推定する場合でも同様だと考えられる。

感情識別は、4 コマ漫画ストーリーデータセットのようなメタ情報を含んだデータセットを用いなければできない研究であるが、ベースラインを超える結果を得ることができたことから、人工知能を用いて漫画の登場キャラクターの感情を推定することが十分に可能であるということを示すことができた。

表 5.3: 実験結果

不均衡データなので、ベースラインは $101/152 \approx 0.664$ である。どのモデルもベースラインを超える精度が出せたことがわかる。

モデル	訓練データ Accuracy	テストデータ		
		Accuracy	Precision (ニュートラル)	Precision (その他)
事前情報なしモデル	1.000	0.693	0.620	0.711
1 事前情報モデル	0.917	0.767	0.608	0.804
2 事前情報モデル	0.916	0.733	0.677	0.792
3 事前情報モデル	0.907	0.700	0.530	0.769

6 まとめと今後の課題

座標情報だけでは推定できない順序を識別するために、文脈から順序を識別するモデルを提案した。データは Manga 109 から「OL ランチ」を用いた。モデルの評価は、正しい順序と誤った順序の 4 コマ漫画の中からを選択できるかどうかとして実験した。

日本語の 4 コマ漫画に対して、言語情報のみを用いて順序推定した例はなく、実験結果の良し悪しを問うことは困難であるが、正しい順序と間違った順序 1 つずつを比較した場合には 69.8 % の精度で識別することができた。その他にも、不完全な順序との比較や 24 通りすべての順序と比較をし、ベースラインを越すことが確認できた。

4 コマ漫画の順序推定の予備実験的に行った 2 コマの順序推定は、推定するコマ数の少なさが必ずしも難易度の緩和につながるわけではないことを示唆する結果となった。コマ数が少なすぎると、セリフがないようなコマを含む 2 コマの推定が不可能であったり、2 コマ単位では置換可能であったりする場合が存在するため識別が困難であり、事前情報としていくつかのコマを用いた方が順序推定をしやすいという、今後の順序推定を行う場合の重要な知見を得ることができた。

また、実験 1.2 と 実験 1.3 より、不均衡データの学習は難しい一方で、4 コマ漫画の正しい順序の選択率は不均衡データを用いて学習をしたほうが良かった。ここから、データ数を削減して不均衡さをなくすよりも、データ数が多い状態で学習をしたほうが良いことがわかった。

追加実験的に行った感情推定では、4 コマ漫画ストーリーデータセットを用いて、人工知能を用いることで漫画の登場キャラクターの感情を推定できる可能性を示すことができた。

今後の展望としては、現在のモデルを更に改良し、言語情報のみでの識別精度を上げることや、画像処理や言語と画像を用いたマルチモーダルなモデル、最終目標である座標情報だけでは推定できない順序推定のための座標情報と組み合わせた順序推定モデルの構築が挙げられる。

謝辞

本研究に対し査読に加え，御助言をいただいた藤本 典幸教授と吉岡 理文教授に厚く御礼申し上げます．また，本研究を進めるにあたり，日々の研究から学会発表に至るまであらゆるご指導ご鞭撻いただいた森 直樹教授と岡田 真助教に深く感謝いたします．

2020年2月28日

参考文献

- [1] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
- [2] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.
- [3] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *CoRR*, abs/1506.06726, 2015.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.
- [5] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
- [6] 石井, 渡辺. マンガからの自動人物検出と識別に関する一検討. 画像電子学会誌, 42(4):457–465, 2013.
- [7] C. Rigaud, N. L. Thanh, J. Burie, J. Ogier, M. Iwata, E. Imazu, and K. Kise. Speech balloon and speaker association for comics and manga understanding. In *ICDAR*, pp. 351–355. IEEE Computer Society, 2015.
- [8] C. Rigaud, S. Pal, J.-C. Burie, and J.-M. Ogier. Toward speech text recognition for comic books. In *Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding*, MANPU ’16, pp. 8:1–8:6, New York, NY, USA, 2016. ACM.
- [9] S. Fujino, N. Mori, and K. Matsumoto. Recognizing the order of four-scene comics by evolutionary deep learning. In *Distributed Computing and Artificial Intelligence, 15th International Conference, DCAI 2018, Toledo, Spain, 20-22 June 2018*, pp. 136–144, 2018.

- [10] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Distributed Representations, pp. 77–109. MIT Press, Cambridge, MA, USA, 1986.
- [11] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.
- [12] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.
- [13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.
- [14] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [15] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.
- [16] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014.
- [17] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

- [18] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016.
- [19] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017.
- [20] T. Ogawa, A. Otsubo, R. Narita, Y. Matsui, T. Yamasaki, and K. Aizawa. Object detection for comics using manga109 annotations. *CoRR*, abs/1803.08670, 2018.
- [21] M. Ueno. Four-scene comic story dataset for softwares on creative process. In *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 17th International Conference SoMeT_18, Granada, Spain, 26-28 September 2018*, pp. 48–56, 2018.
- [22] C. Guérin, C. Rigaud, A. Mercier, F. Ammar-Boudjelal, K. Bertet, A. Bouju, J.-C. Burie, G. Louis, J.-M. Ogier, and A. Revel. ebdtheque: a representative database of comics. In *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1145–1149, 2013.
- [23] 柴田, 河原, 黒橋. Bertによる日本語構文解析の精度向上. 言語処理学会第25回年次大会, 名古屋,(2019.3), pp. 205–208, 2019.
- [24] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50, Valletta, Malta, May 2010. ELRA. <http://is.muni.cz/publication/884893/en>.
- [25] H. Morita, D. Kawahara, and S. Kurohashi. Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015.

- [26] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015.
- [27] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.