補完的物語生成手法に基づき拡張した あらすじ文を用いた映画の興行的成功予測

飯倉 陸
大阪府立大学大学院工学研究科

Riku Iikura
Graduate School of Engineering, Osaka Prefecture University

1 緒言

映画産業は，製作費が数 100 億円以上にも及ぶ大規模なビジネスである．そのため，映画の企画段階で，その映画が興行的に成功するか否かを予測することは，製作陣や製作費の投資家にとってきわめて重要といえる．近年の深層学習技術の発展に伴い，映画の興行収入を予測する手法は複数提案されてきた [1, 2, 3, 4]．その手法の多くは，監督や出演者，公開年などの，映画のメタ情報を特徴量として利用している．一方で，それらメタ情報に，物語の流れを表現したあらすじ文の言語情報を合わせた特徴量を使用する研究は，ほとんど報告されていない．興行的に成功した映画のあらすじ文は，感情的な起伏に富んでいる傾向にあるという研究報告 [5] が存在することから，映画のあらすじ文の言語情報を特徴量として利用することは有効であると考えられる．そこで本研究では，映画のメタ情報と，事前学習済みの Bidirectional Encoder Representation from Transformers（BERT）[6] モデルに基づき処理したあらすじ文の言語情報を合わせたマルチモーダルな特徴量を用いて，映画の興行的成功の予測精度の向上を図る．

映画のあらすじ文データに関して，例えば CMU Movie Summary Corpus [7] などの英語のデータセットは整備されている一方で，日本語のデータセットは未だ十分でない．そのため本研究では，日本語 Wikipedia[1] における映画のあらすじ文データ，一般社団法人日本映画製作者連盟（映連）[2] によって毎年発表される映画の国内興行収入上位作品の情報，および映画レビューサイト Filmarks[3] 上のメタ情報に基づき，映画の興行的成功予測のためのデータセットを新たに構築した．

しかしながら，収集可能な映画のデータ数は限られているため，作成したデータセットのサイズがモデルの予測精度向上のボトルネックになりうる．そこで本研究では，収集したあらすじ文データに対してテキストデータ拡張手法を適用することで，学習モデルの予測精度の向上を図る．テキストデータ拡張手法としてはこれまでに，変形的なデータ拡張 [8, 9] および生成的なデータ拡張 [10, 11] の観点から，複数の手法が提案されている．本研究では，物語的文章の性質に着目した文生成モデルを新たに構築し，それに基づく映画のあらすじ文の生成的な拡張手法を提案する．

例えば，伝統的な映画の脚本構成である三幕構成では，物語の最初と最後が一貫性を保ちながら，感情的な起伏を伴う多様なイベントによって関連付けられている．筆者はこの性質を踏まえ，一貫性のある物語を生成するためには，指定された結末の情報を適切に考慮することが有効であると考えた．そこで本研究では，文章の最初と最後の文から，それらを適切に連結する文章を補完的に生成する，Variational Hierarchical Conversation Recurrent Neural Network（VHCR）[12] ベースの補完的物語生成手法（VHCR-based Complementary Story Generation Model, VHCR-CSGM）[13, 14] を新たに構築する．VHCR は対話生成タスクで提案された Hierarchical Recurrent Encoder Decoder（HRED）[15] と Variational Hierarchical Recurrent Encoder Decoder（VHRED）[16] を発展させた文生成モデルであり，Variational Auto-Encoder（VAE）[17] の構造を応用している．本研究では，VHCR-CSGM に基づき，映画のあらすじ文データを拡張する手法を提案する．従来のテキストデータ拡張手法との比較実験を通じて，提案手法の有効性について検討する．本研究の貢献は以下のとおりである．

（1）日本語の映画あらすじ文およびメタ情報をもつデータセットを新たに作成し，それを用いた映画の興行的成功予測に取り組んだ．

（2）映画のメタ情報とあらすじ文の言語情報の両方を利用したマルチモーダルな手法により，興行的成

[1] https://ja.wikipedia.org/wiki/
[2] http://www.eiren.org/toukei/
[3] https://filmarks.com/

功の予測精度を向上させた．

（3）VHCR-CSGM を用いたテキストデータ拡張手法を提案し，その適用によって学習モデルによる映画の興行的成功の予測精度が向上することを実験的に確認した．

2 関連研究

本章では，（1）映画の興行収入予測，（2）物語の自動生成，および（3）データ拡張の関連研究について，概括する．

2.1 映画の興行収入予測

近年，映画の興行的な成功を予測する機械学習ベースの手法が多数報告されている [1, 2, 3, 4]．Galvão ら [3] は重回帰分析やニューラルネットワーク，決定木を用いて映画の興行収入を推定した．彼らは，その映画が続編であるか否か，MPAA レイティング，ジャンル，製作費，アカデミー賞®[4] などの映画賞の受賞あるいはノミネートの有無，公開時期，監督や俳優，IMDB や Metacritic から収集した批評家の評価などを特徴量として採用した．他にも，Liu ら [18] や Kim ら [19] は Twitter などのソーシャルメディア上の評価を収集して特徴量として用いる手法を提案した．これらの手法は，映画の公開後に得られる興行収入と密接に関係する情報を使用している．しかし，本研究のモチベーションは，映画の企画段階における興行的成功の予測精度向上であるため，映画の公開前に取得可能と考えられる情報のみを特徴量として採用する．

また，Kim ら [5] は CMU Movie Summary Corpus の映画のあらすじデータから，Rotten Tomatoes[5] における評価値の高低を分類するタスクに取り組んだ．彼らは ELMo [20] による文章の埋め込み表現と，NLTK の Vader [21] により算出された各文の感情スコアを結合して，Convolutional Neural Network や Long Short-Term Memory（LSTM）[22] への入力とする手法を提案した．そして実験を通じて，興行的に成功する映画のあらすじ文は感情的な起伏に富んでいる傾向にあることを示した．この研究報告は，筆者が映画の興行的成功の予測モデルへの入力とする特徴量として，あらすじ文の言語情報を付与することが有効であると考える根拠の 1 つである．

[4] https://www.oscars.org/
[5] https://www.rottentomatoes.com/

CMU Movie Summary Corpus ように，英語で書かれた映画のあらすじ文データを対象とした興行収入分析の研究に対して，日本語のあらすじ文データを用いた研究はほとんど存在しない．その理由の 1 つとして，日本語のあらすじ文データを含む映画のオープンデータセットが存在しないことが挙げられる．そのため，本研究では日本語あらすじ文データを用いて興行的成功を予測するためのデータセットを新たに構築した．

上記を踏まえた本研究の特徴は，映画のメタ情報とあらすじ文データの両方を利用する点と，後述する提案手法によりあらすじ文データを拡張してデータセットのサイズ不足を補い，学習モデルの予測精度向上を図る点である．

2.2 物語の自動生成

物語の自動生成は，自然言語処理研究におけるフロンティアであり，これまでに人間による創作物と類似した物語を生成するためのモデルが多数提案されてきた [23, 24]. 近年では，ニューラルネットワーク技術が導入され，物語生成の研究は多様化している．

例えば，計算機による物語の理解と生成のためのタスクとして，入力文に続く物語の適切な結末を選択する Story Cloze Test [25, 26] などが提案されている．このタスクでは特に，機械翻訳や文章要約などの文生成タスクにおいて高い精度を記録した sequence-to-sequence モデル（Seq2Seq） が多く適用される．Roemmele ら [27] は，Recurrent Neural Network（RNN） の一種である LSTM を使用して，与えられたコンテキストに対して適切な最終文を生成するタスクである Story Cloze Test に取り組んだ．また Guan ら [28] は，Story Cloze Test の発展タスクとして，与えられた文章から欠落している文を生成する Story Completion タスクを提案した．理論上，RNN は文章中の次の文字や単語，さらには文の出現確率を予測するように学習する．RNN などのニューラルネットワークに基づいたモデルは一様な文を出力する傾向にあり，独創的な物語の生成が困難であることが問題視される．Gupta ら [29] はこれに対して，コンテキストに存在する重要語に重みを付与することや出現頻度の低い単語の出力を促進することにより，物語の多様な結末を生成する手法を提案した．さらに Mori ら [30] は，実際の文章執筆支援の場面を想定し，補完すべき文の場所を指定する従来の Story Completion タスクに，文が欠落している場所を推定する操作を追加した．

文を連続的に出力して 1 つの物語を生成する研究では，独立した説明文を連結し 1 つの物語を生成するシステム [31] や，物語のシナリオを生成する操作とそれを物語へと変換する操作を組み合わせた階層的な物語生成システム [32] などが提案されている．Martin ら [33] はテキストをイベントのシーケンスに変換し，与えられたイベントからそれに続くイベントを生成する手法（event2event）と，イベントから自然言語文を生成する手法（event2sentence）を提案した．Yao ら [34] は与えられたタイトルあるいはトピックからキーワードベースのプロットやストーリーラインを作成し，それに基づいて物語を生成することでその品質を向上させることを図った．また，物語における主人公の感情の遷移をモデル化し，さらに設定した感情の遷移に基づいて物語を生成する手法 [35] などがある．

上記の物語的文章の生成におけるアプローチと本研究の提案手法の主たる相違点は，物語の最初の文だけでなく，最後の文についても入力として与える点である．さらに VAE の構造を利用することで，語彙的に多様な物語の生成が期待できる．

2.3 データ拡張

データ拡張は，学習モデルの性能を向上させるために訓練データを人工的に生成する，正則化手法の 1 つである．提案されているほとんどの手法は，データの変形または生成に基づくものに大別される．

変形的なデータ拡張では，外部知識が組み込まれた，パラメータ化されていないデータ変換関数に依存して新たなデータポイントが合成される [36]．変形的なデータ拡張手法は，画像認識分野で広く使用されており，例えば訓練画像を回転や移動などの線形変換により拡張することで，訓練モデルの性能が向上する場合がある [37]．

一方，生成的なデータ拡張手法は，潜在変数モデルを利用して訓練データとして妥当なデータサンプルを生成する．VAE や Generative Adversarial Networks [38] などの強力な生成モデルをデータ拡張に活用することの可能性が議論されている．例えば，Visual Question Answering タスク [39] や，一般的な画像分類タスク [40] において，学習モデルの性能向上が報告されている．

自然言語処理におけるデータ拡張では，テキスト中の単語をシソーラスを用いて置換したり，ノイズを混ぜたりすることが一般的である [8]．また，複数の文生成タスクで高い性能を示した GPT-2 [41] を使用する生成的なデータ拡張手法も提案されている [10, 11]．本研究で提案する補完的物語生成手法に基づくデータ拡張手法は，VAE ベースの生成的なデータ拡張手法に分類される．

3 データセット構築

本章では，映画の興行的成功予測のためのデータセットの具体的な構築手順と，データセットに含まれる各特徴量どうしの関係について詳述する．

3.1 構築手順

映連のホームページでは，年間の興行収入が 10 億円以上であった作品が興行収入上位作品として公表されている．それに準拠し，本研究では年間興行収入が 10 億円以上であることを，興行的な成功と定義する．本研究では 2010 年から 2019 年までの 10 年分の発表資料，Wikipedia 上の映画作品記事におけるあらすじ文および Filmarks で管理されている映画の情報を参考に，データセットを構築する．なお，データセットとしての品質を考慮し，MeCab [42] を使用した形態素解析の結果，含まれる単語数が 256 以上のあらすじ文を取得対象とする．取得した映画は 661 本であり，そのうち映連の発表資料に作品名が掲載されていた，つまり，年間の興行収入が 10 億円以上の作品は 255 本であった．一方で，映連の発表資料に作品名が掲載されていなかった作品は 406 本であった．以降では，便宜的に前者を正例，後者を負例と称する．

3.2 各特徴量と興行的成功の関係

表 1 に，データセットに含まれる特徴量を示す．また，図 1 に各量的変数どうしの相関関係を表現するヒートマップを示す．映画の興行的成功はカテゴリ変数として，その他の項目は量的変数として，それぞれの相関比および相関係数を算出した．図 1 に示した各量的変数は，映画の興行的な成功とほとんど相関がないといえる．

一方で，カテゴリ変数としての各映画の監督名と，映画の興行的成功の相関比は 0.91 と強い正の相関を示した．しかし監督名の総数は 464 と訓練用データの総数に対して多く，また 1 人が監督を務めた映画の最大本数は 7 本であったことから，Zero-shot 学習になる可能性が高い．そのため，テスト時の精度向上に大きく寄与するとは考え難い．

表 1: データの特徴量

名称	次元数	説明
success	1	映画が興行的に成功したか否か（正例，負例）をバイナリで表す目的変数.
release year	1	映画の公開年を表す説明変数．2010 年から 2019 年までの公開年について，最小値が 0，最大値が 1 となるように標準化する.
release month	1	映画の公開月を表す説明変数．1 月から 12 月までの公開月について，最小値が 0，最大値が 1 となるように標準化する.
release day	1	映画の公開日を表す説明変数．1 日から 31 日までの公開月について，最小値が 0，最大値が 1 となるように標準化する.
runtime	1	映画の上映時間を表す説明変数．46 分から 209 分までの上映時間について，最小値が 0，最大値が 1 となるように標準化する.
genre	27	映画のジャンルを表す説明変数．“アクション” や “サスペンス”，“アニメ” など 27 種類のジャンルを one-hot 表現する．ジャンルが登録されていない映画が存在し，各映画は最大 3 つのジャンルに属する．ジャンルの組み合わせは 154 通り存在する.
country	38	映画の製作国を表す説明変数．“日本” や “アメリカ”，“イギリス” など 38 の国を one-hot 表現する．製作国が登録されていない映画は存在せず，各映画の製作国は最大 5 カ国とする．製作国の組み合わせは 67 通り存在する.
director	464	映画の監督を表す説明変数．464 人の人物を one-hot 表現する．監督が登録されていない映画が存在し，各映画の監督は最大 1 人である.
actor	3,609	映画の出演者を表す説明変数．3,609 人の人物を one-hot 表現する．出演者が登録されていない映画は存在せず，各映画に対して登録する出演者数は最大 10 人とする.
plot word num	1	映画のあらすじ文に含まれる単語数を表す説明変数．最小値が 0，最大値が 1 となるように標準化する.
title	768	事前学習済み BERT モデルを用いて得られる，タイトルの言語情報を表す 768 次元の分散表現.
plot	768	事前学習済み BERT モデルを用いて，4.1 節に示す手順で取得したあらすじ文の言語情報を表す 768 次元の分散表現.

また，各映画が属するジャンルの組み合わせは 154 通りあった．図 2 に，映画のジャンルの組み合わせと興行的成功の関係を示す．最大 3 つのジャンルの組み合わせと興行的成功の相関比は 0.65 であり，やや強い正の相関を示した．さらに，図 3 に製作国の組み合わせと興行的成功の関係を示す．最大 5 カ国の製作国の組み合わせは 67 通り存在し，興行的成功との相関比は 0.34 であった.

図 4 に，公開年ごとの正例および負例の数を示す．公開年と興行的成功にほとんど相関がないことを踏まえ，2010 年から 2014 年までに公開された映画を訓練および検証用データ，2015 年から 2019 年までに公開された映画をテスト用データとして使用する．表 2 に，作成したデータセットの統計情報を示す．なお，あらすじ文を文単位に分割するために，自然言語処理フレームワークである spaCy[6] に基づき実装された GiNZA [43, 44] を採用した．訓練，検証用データと，テスト用データの性質に大きな差異はないといえる.

[6]https://spacy.io/

4 映画の興行的成功予測モデル

本章では，提案手法である映画の興行的成功予測モデルについて詳述する.

4.1 特徴量としてのテキストデータの処理

映画のタイトル名およびあらすじ文のテキストデータ特徴量として使用するために，事前学習済みの BERT モデル（cl-tohoku/bert-base-japanese-whole-word-masking[7]）を用いて処理し，分散表現を取得する．BERT は複数の双方向 Transformer [45] に基づく汎用言語モデルであり，入力された単語系列および，含まれる各単語に対応する分散表現を出力する.

[7]https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

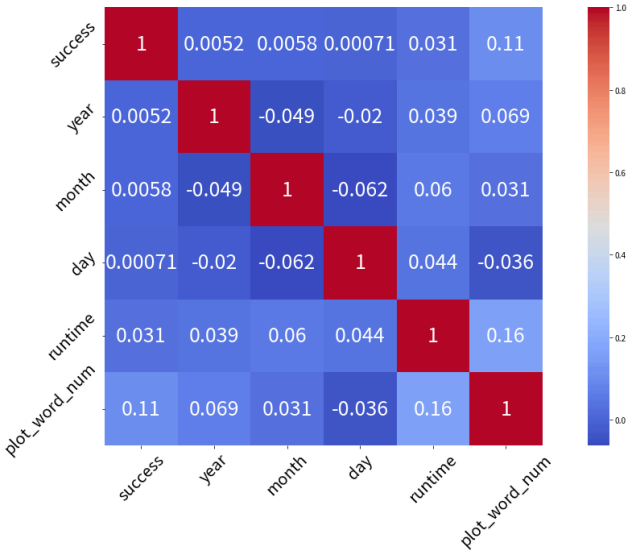


図 1: 各特徴量どうしの相関関係

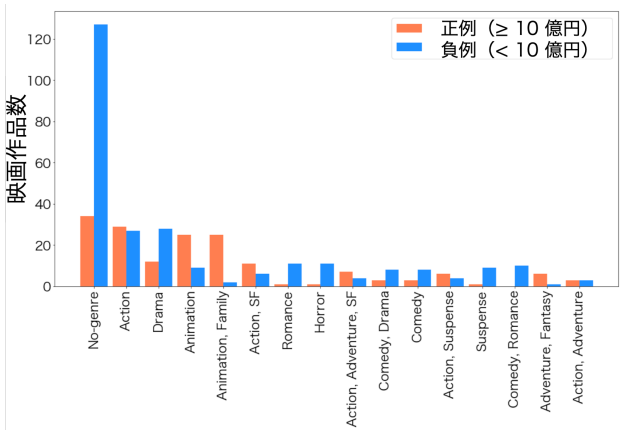


図 2: 映画のジャンルと興行的成功の関係．ただし，‘No-genre’ はジャンル情報がない場合である．

映画のタイトル名を事前学習済み BERT モデルに入力し，入力系列としての前処理で先頭に付与される [CLS] トークンに対する分散表現を，タイトル名に対する分散表現として扱う．

使用する事前学習済み BERT モデルに対する入力系列は，最大長を 512 トークンとする制限が設けられているため，それより多いトークン数が含まれるあらすじ文は前処理段階で先頭 512 トークンのみが抽出される．その場合，あらすじ文の後半に含まれる情報は無視されるため，効果的にあらすじ文を利用できない．本研究では，Pappagari ら [46] に着想を得た手順であらすじ文の分散表現を取得する．図 5，6 に，あらすじ文のチャンク分割の例と分散表現取得の概略図を示す．

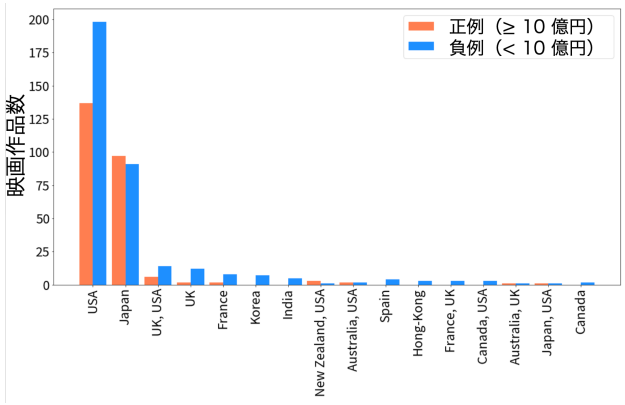


図 3: 映画の製作国と興行的成功の関係

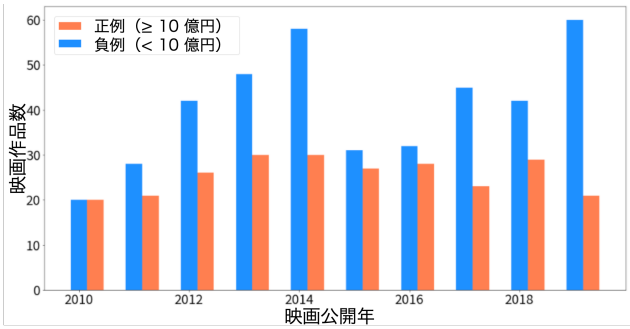


図 4: 映画の公開年と興行的成功の関係

具体的には，あらすじ文の前方から 50 トークンが重複するように 200 トークンごとのチャンクに分割し，それぞれのチャンクの先頭に付与される [CLS] トークンに対する分散表現を取得する．取得した各チャンクに対する分散表現の平均を，対象のあらすじ文の分散表現とする．また，タイトル名とあらすじ文を合わせて分散表現を取得する場合は，タイトル名を最初のチャンクとみなす．

4.2 特徴量とモデルのアーキテクチャ

本研究で検討した，映画の興行的成功予測のための特徴量の組み合わせは以下の通りである．

meta-only: 公開年月日（‘release year’, ‘release month’, ‘release day’），上映時間（‘runtime’），ジャンル（‘genre’），製作国（‘country’），監督（‘director’），出演者（‘actor’）に関する情報（メタ情報）を表す 4,142 次元の特徴量．

plot-only: あらすじ文に対する 768 次元の分散表現．

表 2: データセットの統計情報

	訓練および検証	テスト
データ総数	323	338
正例数	127	128
負例数	196	210
平均文数	25.1	27.4
最小文数	7	6
最大文数	94	156
平均単語数	752.8	817.1
最小単語数	257	259
最大単語数	3,026	3,536

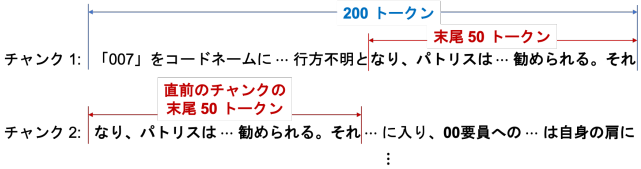


図 5: あらすじ文のチャンク分割例
（Wikipedia: 「007 スカイフォール」[8] より）

meta+plot: メタ情報を表す 4,142 次元の特徴量と，あらすじ文に対する 768 次元の分散表現.

meta+plot word num+title: メタ情報とあらすじ文に含まれる単語数を表す 4,143 次元の特徴量と，タイトル名に対する 768 次元の分散表現.

meta+title+plot: メタ情報を表す 4,142 次元の特徴量と，タイトル名およびあらすじ文に対する 768 次元の分散表現.

meta+plot word num+title+plot: メタ情報とあらすじ文に含まれる単語数を表す 4,143 次元の特徴量と，タイトル名およびあらすじ文に対する 768 次元の分散表現.

上記の特徴量を，分類器としての 3 層の Multi Layer Perceptron（MLP），Random Forest（RF）[47]，XGBoost（XGB）[48], および LightGBM（LGBM）[49] への入力とし，映画の興行的成功（‘success’）を 2 値分類予測する.

メタ情報に対する特徴量とテキストデータに対する分散表現は，異なるモダリティから得られる情報であるため，マルチモーダル学習の側面を備える．伝統的

[8] https://ja.wikipedia.org/wiki/007_スカイフォール

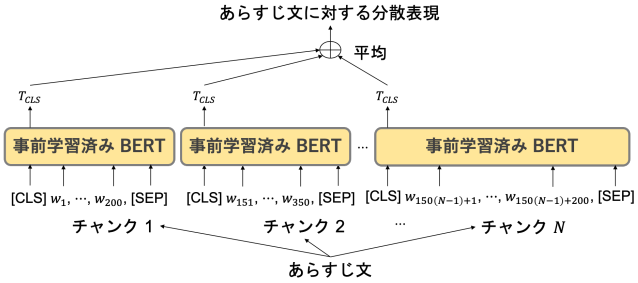


図 6: あらすじ文に対する分散表現取得の概略図

なマルチモーダル学習手法は，Early Fusion と Late Fusion に分類される．前者は異なるモダリティから得られた特徴量を連結し，後者は異なるそれぞれのモダリティに対する個別の分類結果を組み合わせて分類結果を決定する手法である．また，深層学習に基づく分類では，ニューラルネットの隠れ層を連結する手法が一般的に使用される [50, 51].

分類器への入力を，メタ情報 i とテキスト情報 t からなる $x = \{i, t\} \in X$ とし，メタ情報の数値表現を $\gamma(i) \in \mathbb{R}^n$，BERT に基づき取得したテキストに対する分散表現を $\psi(t) \in \mathbb{R}^m$ とする．入力 x と付随するラベル $y \in Y$ が与えられたとき，モデルは最も予測確率の高い y を予測ラベルとして出力する.

$$\hat{y} = \arg\max_{y} P(Y = y \mid x), \tag{1}$$

ここで，$x = i$ あるいは $x = t$ であれば，ユニモーダルな分類器となる.

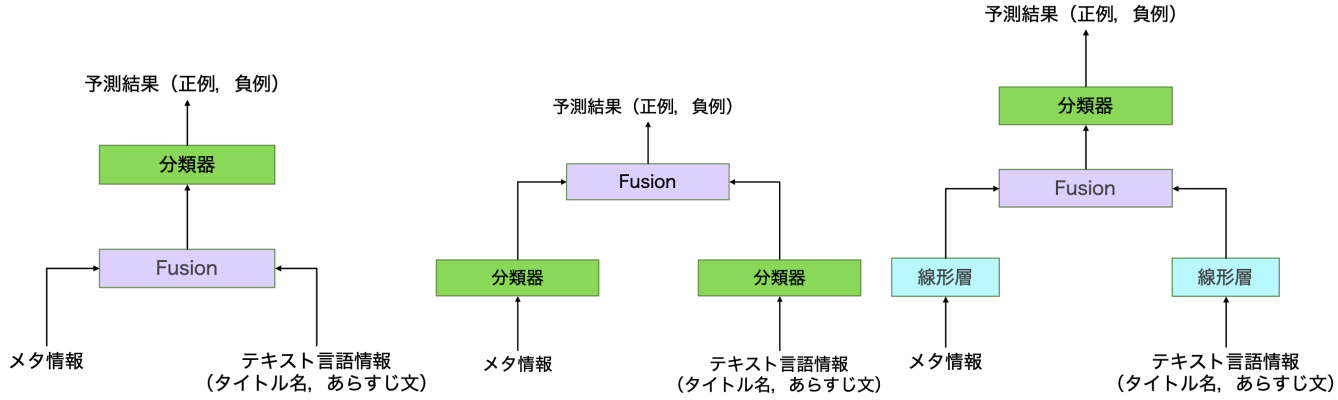
図 7 に，メタ情報とテキストデータを用いた分類における特徴量の結合方法を示す.

Early Fusion: 図 7a に示す Early Fusion は，メタ情報に対する特徴量とテキストデータに対する分散表現を連結し，分類器への入力とする手法である.

$$x' = [\gamma(i); \psi(t)], \tag{2}$$

ここで，$[v_1; v_2]$ はベクトルの連結を表し，$x' \in \mathbb{R}^{(n+m)}$ は連結により得られる新たな特徴ベクトルである.

Late Fusion: また図 7b に示す Late Fusion は，メタ情報とテキスト情報の特徴量をもとにそれぞれ独立に学習した分類器の予測結果を合わせて，最終的な予測結果とする手法である．本研究では，映画の興行的成功の正負に対する各分類器の予測



(a) Early Fusion (b) Late Fusion (c) Joint Fusion

図 7: メタ情報とテキストデータを用いた分類における特徴量の結合

確率の平均値を，最終的な予測確率として用いる．

$$\hat{y} = \arg\max_{y}(\mathrm{mean}(P(Y = y \mid i), P(Y = y \mid t))), \tag{3}$$

ここで，mean(·) は平均値を表す．

Joint Fusion: 図 7c に示す Joint Fusion は，ニューラルネットワークに基づく分類器のための結合手法である．Pooling 処理を適用する場合，ベクトル $\gamma(i) \in \mathbb{R}^n$，$\psi(t) \in \mathbb{R}^m$ の次元数を同一にする必要があるため，4 式，5 式の線形層が追加される．

$$\tilde{\gamma}(i) = \tilde{W}^1\gamma(i) + \tilde{b}^1, \tag{4}$$

$$\tilde{\psi}(t) = \tilde{W}^2\psi(t) + \tilde{b}^2, \tag{5}$$

ここで，$\tilde{\gamma}(i) \in \mathbb{R}^l$, $\tilde{\psi}(t) \in \mathbb{R}^l$, $\tilde{W}^1 \in \mathbb{R}^{n\times l}$, $\tilde{W}^2 \in \mathbb{R}^{m\times l}$, $\tilde{b}^1 \in \mathbb{R}^l$, および $\tilde{b}^2 \in \mathbb{R}^l$ である．

本研究では，6 式に示す線形層の出力を連結する手法（Concatenation）と，7 式に示す線形層の出力に Element-wise Max Pooling 処理を適用する手法（Max Pooling）を検討する．

$$x' = [\tilde{\gamma}(i); \tilde{\psi}(t)], \tag{6}$$

$$x' = \mathrm{pooling}(\tilde{\gamma}(i), \tilde{\psi}(t)). \tag{7}$$

5 補完的物語生成手法

本章では，技術的背景である Recurrent Neural Network Language Model (RNNLM) [52] , HRED, VAE, VHRED，および VHCR について概説したのち，本研究の提案手法である VHCR-CSGM について述べる．

5.1 RNNLM

パラメータ θ を有する RNN は，出力全体の確率分布を分解することにより，可変長のシーケンス $\{w_1, \ldots, w_M\}$ をモデリングする．

$$p_\theta(w_1, \ldots, w_M) = \prod_{m=2}^{M} p_\theta(w_m \mid w_1, \ldots, w_{m-1})p(w_1). \tag{8}$$

モデルはそれぞれの観測値を再帰的に処理する．各タイムステップで，モデルは要素を観察し，以下の式のように内部の隠れ状態を更新する．

$$h_m = f_\theta(h_{m-1}, w_m), \tag{9}$$

ここで，$h_m \in \mathbb{R}^{d_h}$ は，隠れ状態と称され，位置 m までに観察されるトークンのベクトル表現として機能する．特に，最終隠れ状態 h_M は，すべてのトークンの順序に依存する隠れ状態の要約とみなすことが可能である．また，関数 f_θ は双曲線正接関数，LSTM や GRU [53] のゲート付き回帰ユニットなどの活性化関数と呼ばれる非線形関数である．言語モデリングのタスクでは，h_m でエンコードされたコンテキスト情報を使用して，文における次のトークンを予測する．

$$p_\theta(w_{m+1} \mid w_1, \ldots, w_m) = p_\theta(w_{m+1} \mid h_m). \tag{10}$$

出力が語彙 V 内に存在すると仮定するとき，RNNLM [52] は，隠れ状態 h_m のアフィン変換に適用されるソフトマックス関数を用いて，出力分布をパラメータ化する．モデルのパラメータは，最急降下法を使用して対数尤度を最大化することで学習される．

5.2 HRED

HRED [15] は，RNNLM の拡張手法であり，エンコーダ・デコーダモデルの構造を対話生成のために拡張したものである．HRED は，各出力系列がサブシーケンスのシーケンスと，トークンのサブシーケンスという 2 段の階層にモデル化可能であることを前提としている．文章は文というサブシーケンスが連なるシーケンスであり，各文は単語が連なるシーケンスとして表現できる．

本研究では， 1 つのあらすじ文 T を，N 個の文からなる系列 $T = \{S_1, \ldots, S_N\}$ として考える．S_n はそれぞれ M_n 個のトークンで構成されており，$S_n = \{w_{n,1}, \ldots, w_{n,M_n}\}$ である．ここで，$w_{n,m}$ は語彙 V に基づき，文 S_n における位置 m のトークンを表現する．文章生成モデルは，生成されうる任意のあらすじ文に対して，パラメータ θ により制御される確率分布 p_θ をパラメータ化する．T の確率分布は以下の 11 式のように表される．

$$\begin{aligned} p_\theta(S_1, \ldots, S_N) &= \prod_{n=1}^{N} p_\theta(S_n \mid S_{<n}), \\ &= \prod_{n=1}^{N} \prod_{m=1}^{M_n} p_\theta(w_{n,m} \mid S_{<n}, w_{n,<m}), \end{aligned} \tag{11}$$

ただし, $S_{<n} = \{S_1, \ldots, S_{n-1}\}$, $w_{n,<m} = \{w_{n,1}, \ldots, w_{n,m-1}\}$ であり，これらは S_n より前にあるトークンである．

図 8 に, HRED のグラフィカルモデルを示す．HRED の最大の特徴は, エンコーダ RNN, コンテキスト RNN, デコーダ RNN という 3 つの RNN モジュールで構成されている点である．本研究で扱うタスクにおいて，あらすじ文における各文は，エンコーダ RNN f_θ^{enc} によって実数値のベクトルに決定論的にエンコードされる．ここで，文ベクトルは文に含まれる最後のトークンが処理されたのちに取得される隠れ状態である．

$$h_{n-1}^{\mathrm{enc}} = f_\theta^{\mathrm{enc}}(S_{n-1}). \tag{12}$$

エンコーダ RNN の最終状態は 1 段上の階層であるコンテキスト RNN f_θ^{ctx} に入力として与えられる．コンテキスト RNN は内部の隠れ状態を更新してその時点までのすべての情報を反映する．文 S_n を処理したのち, 時刻 n までの文情報を要約したコンテキスト RNN の隠れ状態は，次の文 S_{n+1} を予測するために使用される．

$$h_n^{\mathrm{ctx}} = f_\theta^{\mathrm{ctx}}(h_{n-1}^{\mathrm{ctx}}, h_{n-1}^{\mathrm{enc}}). \tag{13}$$

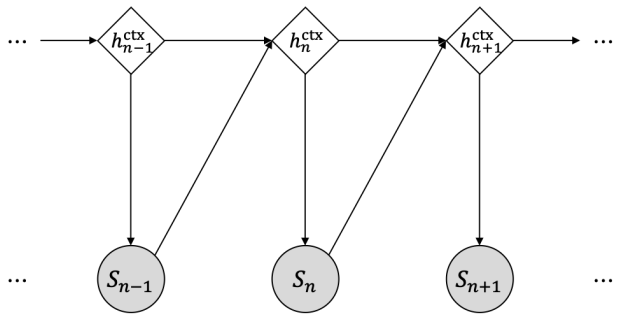


図 8: HRED のグラフィカルモデル

デコーダ RNN f_θ^{dec} は，コンテキスト RNN の隠れ状態を取得して，次の発話のトークン全体における確率分布を生成することで，次の文を予測する．デコーダ RNN と RNNLM の重要な相違点は, 次の文予測においてコンテキスト RNN の隠れ状態を利用する点である．

$$p_\theta(S_n \mid S_{<n}) = f_\theta^{\mathrm{dec}}(S \mid h_n^{\mathrm{ctx}}). \tag{14}$$

本研究では，エンコーダ，コンテキスト，およびデコーダ RNN のために GRU を採用し，他の場所では活性化関数として双曲線正接関数を採用する．

5.3 VAE

VAE [17] とは，変分ベイズ法を応用したモデルである．VAE では，事前分布 $p(\boldsymbol{z})$ からサンプリングされる潜在変数 $\boldsymbol{z}$ から，データポイント $\boldsymbol{x}$ が生成される．ここで，θ をパラメータとする条件付き確率分布 $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ を仮定し，対数周辺尤度 $\log p_\theta(\boldsymbol{x})$ の最大化を目的とする．しかし，$\log p_\theta(\boldsymbol{x})$ を直接的に最大化することは困難であるため，認識モデル $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ により $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$ を近似し，変分下界の最大化を試みる．

$$\begin{aligned} \log p_\theta(\boldsymbol{x}) &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) \\ &= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right] - D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z})\right), \end{aligned} \tag{15}$$

ここで，$D_{\mathrm{KL}}(\cdot \,||\, \cdot)$ は，カルバック・ライブラー・ダイバージェンス（KL 情報量）を表す．

5.4 VHRED

VHRED [16] は，対話モデリングのために提案された，発話ごとの潜在変数を HRED に導入した文生成モデルである．図 9 に，VHRED のグラフィカルモ

デルを示す．本研究のあらすじ文生成に適用する場合，あらすじ文における過去の文の系列 $S_1, \ldots, S_{n-1}$ が与えられたとき，VHRED は続く文 S_n を次のように生成する．時刻 n において，エンコーダ RNN f_θ^{enc} は文 S_{n-1} を入力として受け取り，エンコードベクトル h_{n-1}^{enc} を出力する．

$$h_{n-1}^{\mathrm{enc}} = f_\theta^{\mathrm{enc}}(S_{n-1}). \tag{16}$$

コンテキスト RNN f_θ^{ctx} は，エンコードベクトルを用いて隠れ状態を更新しながら，あらすじ文のコンテキストをモデリングする．

$$h_n^{\mathrm{ctx}} = f_\theta^{\mathrm{ctx}}(h_{n-1}^{\mathrm{ctx}}, h_{n-1}^{\mathrm{enc}}). \tag{17}$$

コンテキストベクトル h_n^{ctx} は，条件付き確率 $p_\theta(\boldsymbol{z}_n^{\mathrm{snt}} \mid S_{<n})$ を定義する．これは，平均 μ_n と対角分散 σ_n がフィードフォワードニューラルネットワークによって与えられる，分解されたガウス分布である．

$$\begin{aligned} p_\theta(z_n^{\mathrm{snt}} \mid S_{<n}) &= \mathcal{N}(z \mid \mu_n, \sigma_n \boldsymbol{I}), \\ \text{where } \mu_n &= \mathrm{MLP}_\theta(h_n^{\mathrm{ctx}}), \\ \sigma_n &= \mathrm{Softplus}(\mathrm{MLP}_\theta(h_n^{\mathrm{ctx}})). \end{aligned} \tag{18}$$

最終的に，デコーダ RNN f_θ^{dec} は，コンテキストベクトル h_n^{ctx} と潜在変数 z_n^{snt} を条件として文 S_n を生成する．

$$p_\theta(S_n \mid S_{<n}) = f_\theta^{\mathrm{dec}}(S \mid h_n^{\mathrm{ctx}}, z_n^{\mathrm{snt}}). \tag{19}$$

変分事後分布は，分解されたガウス分布であり，平均と対角分散は，ターゲットとなる文とコンテキストから次のように予測される．

$$\begin{aligned} q_\phi(z_n^{\mathrm{snt}} \mid S_{\leq n}) &= \mathcal{N}(z \mid \mu'_n, \sigma'_n \boldsymbol{I}), \\ \text{where } \mu'_n &= \mathrm{MLP}_\phi(S_n, h_n^{\mathrm{ctx}}), \\ \sigma'_n &= \mathrm{Softplus}(\mathrm{MLP}_\phi(S_n, h_n^{\mathrm{ctx}})). \end{aligned} \tag{20}$$

5.5 VHCR

VHCR [12] は, 対話生成タスクで提案された VHRED の発展モデルであり, 階層的かつ回帰的な VAE モデルを用いて対話の全体的な特徴と個々の発話をモデル化する．図 10 に VHCR のグラフィカルモデルを示す．VHCR は，VHRED で導入された各ステップでの発話のエンコードと生成に使用されるローカルレベルの潜在変数に加えて，より複雑な対話構造をエンコードするためのグローバルレベルの潜在変数を導入する．

$$h_{n-1}^{\mathrm{enc}} = f_\theta^{\mathrm{enc}}(S_{n-1}). \tag{21}$$

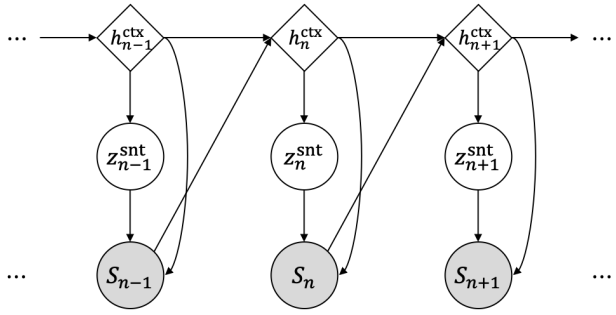


図 9: VHRED のグラフィカルモデル

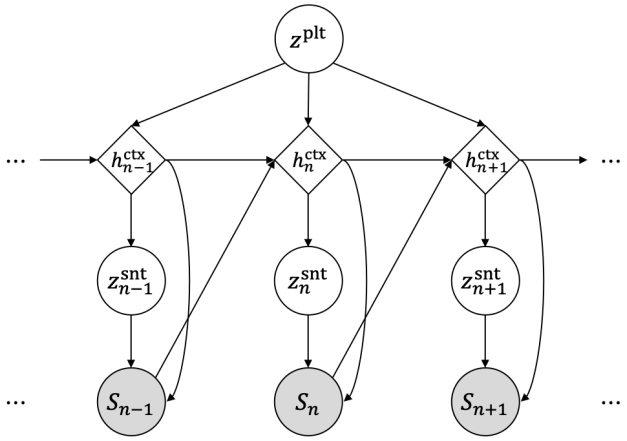


図 10: VHCR のグラフィカルモデル

$$h_n^{\mathrm{ctx}} = \begin{cases} \mathrm{MLP}_\theta(z^{\mathrm{plt}}), & \text{if } n = 0, \\ f_\theta^{\mathrm{ctx}}(h_{n-1}^{\mathrm{ctx}}, h_{n-1}^{\mathrm{enc}}, z^{\mathrm{plt}}), & \text{otherwise.} \end{cases} \tag{22}$$

$$p_\theta(S_n \mid S_{<n}, z_n^{\mathrm{snt}}, z^{\mathrm{plt}}) = f_\theta^{\mathrm{dec}}(S \mid h_n^{\mathrm{ctx}}, z_n^{\mathrm{snt}}, z^{\mathrm{plt}}). \tag{23}$$

$$p_\theta(z^{\mathrm{plt}}) = \mathcal{N}(z \mid \boldsymbol{0}, \boldsymbol{I}). \tag{24}$$

$$\begin{aligned} p_\theta(z_n^{\mathrm{snt}} \mid S_{<n}, z^{\mathrm{plt}}) &= \mathcal{N}(z \mid \mu_n, \sigma_n \boldsymbol{I}), \\ \text{where } \mu_n &= \mathrm{MLP}_\theta(h_n^{\mathrm{ctx}}, z^{\mathrm{plt}}), \\ \sigma_n &= \mathrm{Softplus}(\mathrm{MLP}_\theta(h_n^{\mathrm{ctx}}, z^{\mathrm{plt}})). \end{aligned} \tag{25}$$

本研究における，あらすじ文全体を制御する潜在変数 z^{plt} を推論するために，エンコーダ RNN により生成された文ベクトルに対して，双方向 RNN f^{plt} を用いる．

$$\begin{aligned} q_\phi(z^{\mathrm{plt}} \mid S_{\leq n}) &= \mathcal{N}(z \mid \mu_n^{\mathrm{plt}}, \sigma_n^{\mathrm{plt}} \boldsymbol{I}), \\ \text{where } h^{\mathrm{plt}} &= f^{\mathrm{plt}}(h_1^{\mathrm{enc}}, \ldots, h_n^{\mathrm{enc}}), \\ \mu_n^{\mathrm{plt}} &= \mathrm{MLP}_\phi(h^{\mathrm{plt}}), \\ \sigma_n^{\mathrm{plt}} &= \mathrm{Softplus}(\mathrm{MLP}_\phi(h^{\mathrm{plt}})). \end{aligned} \tag{26}$$

ローカル潜在変数 z^{snt} は，グローバル潜在変数 z^{plt} とそれ以前に観測された文情報に条件付きで依存する．

$$
\begin{aligned}
q_\phi(z_n^{\mathrm{snt}} \mid S_{\leq n}, z^{\mathrm{plt}}) &= \mathcal{N}(z \mid \mu'_n, \sigma'_n \boldsymbol{I}), \\
\text{where } \mu'_n &= \mathrm{MLP}_\phi(S_n, h_n^{\mathrm{ctx}}, z^{\mathrm{plt}}), \\
\sigma'_n &= \mathrm{Softplus}(\mathrm{MLP}_\phi(S_n, h_n^{\mathrm{ctx}}, z^{\mathrm{plt}})).
\end{aligned} \tag{27}
$$

5.6 VHCR-CSGM

本研究で提案する VHCR-CSGM は，対象とするあらすじ文の最初と最後の文を入力とし，その間の文章を補完的に生成することにより 1 つの物語を完成させる．図 11 に，VHCR をベースとする CSGM のグラフィカルモデルを示す．図 11 中の h_n^{ctx} は時刻 $n-1$ 以前の文の情報をエンコードするコンテキスト，z_n^{snt} は各文レベルの潜在変数，z^{plt} はあらすじ全体のコンテキストを提供するグローバルな潜在変数である．文 S_n は，h_n^{ctx}, z_n^{snt}, z^{plt} を条件として生成される．時刻 n における文をエンコードするためのエンコーダ RNN $f_\theta^{\text{p-enc}}$ と，最終文 S_N をエンコードするためのエンコーダ RNN $f_\theta^{\text{l-enc}}$ を使用する．それぞれの最終隠れ状態を Element-wise MaxPooling により処理することで，時刻 n におけるコンテキスト RNN f_θ^{ctx} のための入力を得る．

$$
\begin{aligned}
h_{n-1}^{\text{p-enc}} &= f_\theta^{\text{p-enc}}(S_{n-1}), \\
h_N^{\text{l-enc}} &= f_\theta^{\text{l-enc}}(S_N), \\
h_{n-1}^{\mathrm{enc}} &= \mathrm{pooling}(h_{n-1}^{\text{p-enc}}, h_N^{\text{l-enc}}).
\end{aligned} \tag{28}
$$

以降の処理は，VHCR における 22-27 式の処理に従う．

本研究におけるデータ拡張のためのあらすじ文生成では，元のあらすじ文に付随するラベル情報に即した文章を生成するために，入力とするあらすじ文の最初の文の先頭にラベルを付与する．

$$
S_1 = \{y, [\texttt{SEP}], w_{1,1}, w_{1,2}, \ldots, w_{1,M_1}\}, \tag{29}
$$

ここで，y はあらすじ文 T を持つデータサンプルに付随するラベル，[SEP] はラベルと単語系列を分割するための特別なトークンをそれぞれ表す．

6 数値実験

本章ではまず，予測において使用する，特徴量の組み合わせおよびモデルのアーキテクチャに関する比較実験について述べる．続いて，VHCR-CSGM および従来のテキストデータ拡張手法を用いて拡張したデータに基づく学習モデルのタスク解答性能について確認する．

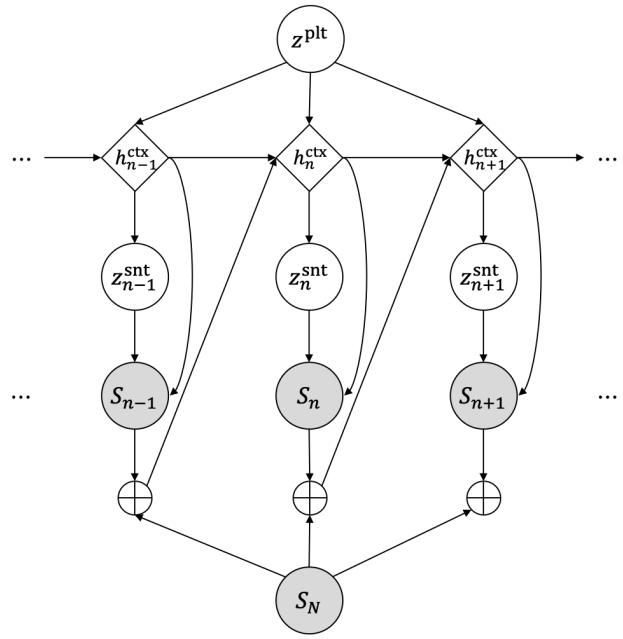


図 11: VHCR-CSGM のグラフィカルモデル

6.1 数値実験 1: 特徴量と分類器の比較

6.1.1 手法

4.2 節で示したデータ特徴量と分類器に加えて，あらすじ文のみを入力として転移学習した BERT モデルをベースライン手法とする．前述のとおり，使用する事前学習済み BERT モデルでは，入力系列の最大長を 512 とする制限が設けられている．この制限に対して Sun ら [54] は，重要な情報は文章の最初と最後に書かれる場合が多いことに着目し，対象の文章の先頭と末尾を連結した入力系列を使用した．Sun らは，IMDB におけるレビュー文の 2 値分類タスク [55] と，中国語の Sogou ニュースコーパス [54, 56] のカテゴリ分類タスクにおいて，その有効性を確認した．上記を踏まえ，本実験では次の入力系列を使用する．

head-only: あらすじ文の先頭から 512 トークンを入力系列とする．

head+tail: あらすじ文の先頭から 256 トークン，末尾から 256 トークンをそれぞれ切り出し，連結して 512 トークンの入力系列とする．

6.1.2 ハイパーパラメータ

4.2 節で示したデータ特徴量と分類器を用いて，表 2 に示したデータセットにおける映画の興行的成功（‘success’）を 2 値分類する．実験は 10 分割交差検証に基づき，試行ごとに各モデルのハイパーパラメータを Optuna [57] によって自動調節する．本実験では，使用する RF，XGB，LGBM，および MLP を，それぞれ Python のオープンソースライブラリである scikit-learn[9], XGBoost Python-package[10], LightGBM Python-package[11]，および Pytorch[12] に基づいて実装する．表 3 に，探索対象としたハイパーパラメータ名とその探索範囲を示す．Optuna の探索回数はそれぞれ 100 回とし，表 3 に記載されていないハイパーパラメータについては，それぞれデフォルトの値を使用する．なお，BERT の転移学習では，全層の重みを更新する実験時間の都合上，Optuna によるパラメータ探索回数を 10 回とする．各試行で検証データに対して最も高い評価値（F1 値）が得られたモデルを用いてテストデータのラベルを予測する．

6.1.3 評価指標

本実験では，評価指標として Accuracy と F1 値を使用する．また本研究では，国内の年間興行収入が 10 億円以上であることを，映画の興行的な成功と定義した．表 4 に，正例映画の興行収入の統計情報を示す．さらに図 12，13 に，正例映画の公開年およびあらすじ文の単語数と，興行収入の関係をそれぞれ示す．正例映画の公開年と興行収入の相関係数は 0.164，あらすじ文の単語数と興行収入の相関係数は 0.235 である．

正例映画においても興行収入には差があり，例えば，興行収入が 10 億円の作品と，100 億円の作品では予測の難易度に差があることが考えられる．そこで，正解ラベルが正例である映画について，その予測結果と興行収入の実数値の関係を確認するために，負例と予測された正例の映画の興行収入の平均値（Mean Wrong Revenue, MWR）を 30 式のように定義する．

$$\mathrm{MWR} = \frac{1}{N^{\mathrm{pos}}} \sum_{i=1}^{N^{\mathrm{pos}}} r_i (1 - \hat{y}_i^{\mathrm{pos}}), \tag{30}$$

ここで，正例および負例ラベルはそれぞれ 1, 0 の値とする．また，N^{pos} は正例の総数，r は正例の映画の興

[9] https://github.com/scikit-learn/scikit-learn
[10] https://github.com/dmlc/xgboost
[11] https://github.com/microsoft/LightGBM
[12] https://github.com/pytorch/pytorch

表 3: 各モデルのハイパーパラメータの値あるいは探索範囲.

分類器	パラメータ名	値あるいは探索範囲
BERT fine-tuning	Optimizer	Adam
	Learning rate	$[1 \times 10^{-5}, 1 \times 10^{-4}]$
	Loss function	Cross Entropy Loss
	Train epochs	[5, 20]
	Train batch size	{16, 32}
RF	n_estimators	{2, 4, 8, 16, 32, 64, 128, 256}
	max_depth	[1, 128]
	min_samples_split	[2, 16]
	min_samples_leaf	[1, 8]
XGB	booster	{“gbtree”, “gblinear”, “dart”}
	lambda	$[1 \times 10^{-8}, 1.0]$
	alpha	$[1 \times 10^{-8}, 1.0]$
	subsample	[0.2, 1.0]
	colsample_bytree	[0.2, 1.0]
	max_depth	[3, 9]
	min_child_weight	[2, 10]
	eta	$[1 \times 10^{-8}, 1.0]$
	gamma	$[1 \times 10^{-8}, 1.0]$
	grow_policy	{“depthwise”, “lossguide”}
	sample_type	{“uniform”, “weighted”}
	normalize_type	{“tree”, “forest”}
	rate_drop	$[1 \times 10^{-8}, 1.0]$
	skip_drop	$[1 \times 10^{-8}, 1.0]$
LGBM	boosting_type	“gbdt”
	learning_rate	$[1 \times 10^{-8}, 1 \times 10^{-1}]$
	lambda_l1	$[1 \times 10^{-8}, 10]$
	lambda_l2	$[1 \times 10^{-8}, 10]$
	num_leaves	[2, 256]
	feature_fraction	[0.4, 1.0]
	bagging_fraction	[0.4, 1.0]
	bagging_freq	[1, 8]
	min_child_samples	[5, 100]
MLP	Optimizer	Adam
	Learning rate	$[1 \times 10^{-8}, 1 \times 10^{-4}]$
	Loss function	Cross Entropy Loss
	Hidden unit	{128, 256, 512, 1024, 2048}
	Train epochs	[30, 50]
	Train batch size	{4, 8, 16, 32}

表 4: 正例映画の興行収入の統計情報

[単位: 億円]

	訓練および検証	テスト
平均（標準偏差）	29.70 (29.2)	38.80 (34.0)
最小値	10.0	10.0
25%	14.2	15.55
中央値	20.2	27.90
75%	33.0	50.725
最大値	255.0	250.3

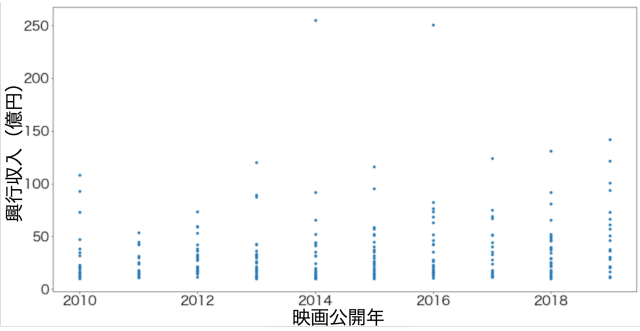


図 12: 正例映画の公開年と興行収入の関係

行収入，および $\hat{y}^{\text{pos}}$ は正例に対する予測ラベルを表す．この MWR の値が低いほど，より興行収入の高い正例映画のラベルを正しく予測したといえる．すべての正例を正しく予測した場合に MWR は最小値 0 となる．一方，テストデータにおけるすべての正例を誤って予測した場合に MWR は最大値となり，その値はすべての正例の興行収入の平均値である 38.8 である．

6.1.4 結果と考察

表 5 に，各特徴量と分類器による予測結果の評価値を示す．ここで，Random モデルは無作為に予測ラベルを生成した場合であり，結果は平均（標準偏差）で表される．また，ボールド体の数値は，各分類器を使用した場合に得られた各評価指標の最高値を表す．まず，ベースライン手法である，映画のあらすじ文のみを入力として転移学習した BERT による予測では，入力系列を head+tail とした場合の方が，head-only とした場合よりも Accuracy と F1 値は高くなった．そして，検討したすべての特徴量と分類器の組み合わせについて比較すると，特徴量 meta+plot word num+title+plot を Early Fusion によって結合した XGB によって予測した場合に最も高い Accuracy と F1 値が得られた．

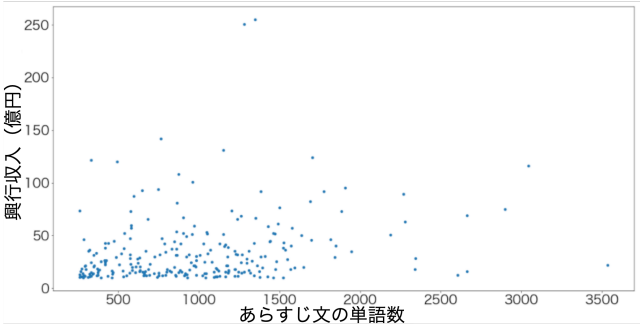


図 13: 正例映画のあらすじ文の単語数と興行収入の関係

特徴量の組み合わせの比較: 表 5 より，メタ情報のみを特徴量とした場合と，メタ情報とあらすじ文に対する分散表現を合わせて特徴量とした場合を比較すると，Late Fusion により特徴量を結合する RF 以外の手法について，後者の方が Accuracy と F1 値が高いことが確認された．また MWR の値が小さくなっていることから，あらすじ文の情報を付与することで，興行収入の高い正例の映画作品について正しく予測可能であるといえる．これらの結果は，映画の興行的成功を予測するために，メタ情報とテキスト情報を合わせたマルチモーダルな特徴量が有効であることを示唆している．さらに，meta+title+plot および meta+plot word num+title に対する評価値より，あらすじ文自体の代わりにあらすじ文の単語数を使用した場合の予測評価値は低かった．このことから，単にあらすじ文の長さだけではなく，言語的な情報が有効に利用され，予測精度が向上したことが考察される．

特徴量の結合手法の比較: 表 5 より，メタ情報とテキスト情報の結合手法として Early Fusion を採用した場合の方が，Late Fusion を採用した場合よりも予測評価値が高くなった．これは，Late Fusion ではメタ情報とテキスト情報に関するそれぞれの分類器を独立に学習させたことで，メタ情報とテキスト情報の相互関係を適切に利用できなかったためであると考えられる．

さらに，学習器として MLP を用いた場合の実験結果について確認すると，特徴量として meta+plot，特徴量の結合手法として Max Pooling を採用した場合の Accuracy と F1 値が最大となった．結合手法のみを比較すると，Early Fusion, Concate-

nation，および Max Pooling のそれぞれを採用した場合の最高値にはほとんど差がなく，優劣は確認されなかった．

6.2 数値実験 2: テキスト拡張手法の比較

6.2.1 手法

提案手法である VHCR-CSGM を含む, テキストデータ拡張手法を用いて映画のあらすじ文を拡張し，それに基づき訓練したモデルの予測性能について確認する．具体的には，テキストデータ拡張手法により，交差検証における訓練データサンプルのあらすじ文からそれぞれ 9 つのあらすじ文を生成し，訓練データサイズを 10 倍に拡張する．なお，生成されたサンプルのメタ情報は元のサンプルと同一である．比較のために使用したテキストデータ拡張手法は以下のとおりである．

1. **Easy Data Augmentation（EDA）[8]:** 以下の 4 つの操作のうち 1 つを無作為に選び実行するデータ拡張手法.

   (1) **Synonym Replacement:** ストップワードでない任意の数の単語を無作為に選択し類義語と置換する．

   (2) **Random Insertion:** ストップワードでない任意の数の単語を無作為に選択し，それらの類義語を文章中に無作為に挿入する．

   (3) **Random Swap:** 任意の数の単語のペアを無作為に選択し, それらの位置を入れ替える．

   (4) **Random Deletion:** 文章中の単語を任意の確率で削除する．

   なお，本研究ではあらすじ文を単語に分割する手法として MeCab を使用する．また, SlothLib で管理されているストップワードリスト[13] と，日本語の概念辞書である日本語 WordNet [58] を使用する．

2. **Conditional BERT（CBERT）[9]:** BERT の事前学習時に対象とする, Masked Language Modeling タスクに着想を得た，文脈とラベル情報に即した単語の置換ベースのテキストデータ拡張手法．従来の Masked Language Modeling タスクのみを学習した BERT モデルは，文脈情報のみに基づき代替となる単語を予測するため，テキストデータに付与されたラベル情報については考慮されない．そこで Wu ら [9] は，データに付与されたラベルとコンテキストの情報からマスクされた単語を予測する条件付き言語モデリング（Conditional Masked Language Modeling）を BERT モデルの転移学習のためのタスクとして設定した．モデルはテキストデータ拡張において，任意の確率でマスクされた文章中の単語を，データに付与されたラベルとコンテキストの情報に基づき，異なる単語に置換する．

3. **GPT-2$_{\text{context}}$ [10, 11]:** GPT-2 [41] に基づくテキストデータ拡張手法．GPT-2 は Transformer に基づくモデルであり，Web 上のテキストコーパスに基づいて事前学習される．使用されるコーパスは，$\mathcal{U} = \{w_1, w_2, \ldots, w_n\}$ で表されるトークン列である．モデルは, $\mathcal{U}$ に対して以下の損失関数を用いて事前訓練される．

$$L(\mathcal{U}) = -\sum_i \log P(w_i \mid w_{i-j}, \ldots, w_{i-1}; \Theta), \tag{31}$$

   ここで，j はコンテキストのウィンドウサイズであり，条件付き確率 P はパラメータ Θ を持つニューラルネットワークを用いてモデリングされる．これらのパラメータは，確率的勾配降下法に基づいて学習される．

   GPT-2$_{\text{context}}$ は，ラベルの情報がコンテキストの一部として使用するために，訓練データセットに含まれるすべての系列を以下のように連結した系列を用いて転移学習される．

$$\mathcal{U}^* = y_1\,\texttt{[SEP]}\,x_1\,\texttt{[EOS]}\,y_2 \cdots y_n\,\texttt{[SEP]}\,x_n\,\texttt{[EOS]} \tag{32}$$

   ここで y, x は，それぞれデータのラベルとテキストを表す．また，[SEP] は文とラベルを区切るための特別なトークンであり，[EOS] は文の終了を意味する特別なトークンである．生成時には y_i [SEP] $w_1 \cdots w_k$ をプロンプトとしてモデルに入力し，[EOS] が生成されるまで出力を続行する．ここで，$w_1 \cdots w_k$ は x_i に含まれる先頭から k 個のトークン列である．本研究では，Kumar ら [11] の設定に従い $k = 3$ とする．また，GPT-2 の事前学習済みモデルとして，日本語 CC-100[14] と日本語 Wikipedia からなる約 75

[13] http://svn.sourceforge.jp/svnroot/slothlib/CSharp/Version1/SlothLib/NLP/Filter/StopWord/word/Japanese.txt

[14] http://data.statmt.org/cc-100/

表 5: 各特徴量と分類器の予測評価値

分類器	結合手法	特徴量	Accuracy (↑)	F1 値 (↑)	MWR (↓)
Random	-	-	49.0 (1.6)	48.2 (1.7)	19.4 (1.8)
BERT fine-tuning (head-only)	-	-	69.9 (1.9)	66.3 (2.1)	17.5 (3.3)
BERT fine-tuning (head+tail)	-	-	**71.5 (3.5)**	**67.9 (5.4)**	**15.4 (3.9)**
RF	Unimodal	meta-only	70.3 (1.7)	62.4 (3.2)	23.4 (3.2)
		plot-only	69.2 (4.4)	65.3 (4.9)	17.5 (3.7)
	Early Fusion	meta+plot	**70.9 (1.9)**	**67.3 (2.3)**	16.7 (2.0)
		meta+plot word num+title	62.3 (3.7)	54.4 (4.0)	26.4 (4.0)
		meta+title+plot	70.1 (3.2)	66.5 (3.3)	16.5 (2.2)
		meta+plot word num+title+plot	69.3 (3.2)	66.2 (3.0)	**15.7 (2.2)**
	Late Fusion	meta+plot	66.9 (1.5)	51.8 (4.1)	31.0 (3.1)
		meta+plot word num+title	65.8 (1.5)	49.7 (4.3)	32.4 (2.5)
		meta+title+plot	66.9 (1.1)	52.1 (2.8)	30.6 (2.0)
		meta+plot word num+title+plot	69.3 (1.7)	57.9 (4.2)	26.8 (3.3)
XGB	Unimodal	meta-only	72.7 (1.5)	66.9 (1.6)	20.3 (3.8)
		plot-only	70.2 (2.5)	66.4 (2.2)	17.1 (2.4)
	Early Fusion	meta+plot	74.0 (2.1)	69.3 (1.9)	17.5 (1.1)
		meta+plot word num+title	69.2 (2.6)	65.0 (2.6)	17.5 (2.5)
		meta+title+plot	74.9 (2.7)	71.1 (2.3)	15.5 (1.4)
		meta+plot word num+title+plot	**75.5 (2.4)**	**71.4 (2.2)**	15.9 (1.8)
	Late Fusion	meta+plot	74.4 (1.1)	70.2 (1.5)	15.6 (1.7)
		meta+plot word num+title	70.6 (2.1)	65.0 (3.3)	20.7 (2.3)
		meta+title+plot	74.0 (1.4)	70.0 (2.1)	**14.8 (2.4)**
		meta+plot word num+title+plot	73.6 (2.3)	69.8 (2.5)	**14.8 (2.5)**
LGBM	Unimodal	meta-only	71.7 (1.0)	67.6 (1.2)	15.7 (1.5)
		plot-only	71.8 (1.6)	68.1 (2.3)	15.8 (2.8)
	Early Fusion	meta+plot	72.0 (1.4)	68.9 (2.1)	14.6 (3.4)
		meta+plot word num+title	69.6 (1.0)	63.9 (2.6)	19.1 (3.6)
		meta+title+plot	73.1 (1.4)	**70.6 (1.6)**	**12.8 (3.3)**
		meta+plot word num+title+plot	72.3 (1.5)	69.3 (1.8)	14.2 (2.9)
	Late Fusion	meta+plot	74.0 (2.4)	69.3 (3.7)	16.4 (3.2)
		meta+plot word num+title	72.1 (1.8)	65.5 (2.3)	19.4 (2.3)
		meta+title+plot	**74.2 (1.8)**	69.8 (2.7)	15.6 (2.4)
		meta+plot word num+title+plot	74.0 (1.8)	70.2 (2.4)	14.6 (2.3)
MLP	Unimodal	meta-only	72.1 (6.2)	67.3 (5.4)	20.2 (4.1)
		plot-only	68.1 (8.2)	65.0 (8.4)	13.9 (6.5)
	Early Fusion	meta+plot	72.2 (6.2)	69.0 (7.7)	**13.8 (4.2)**
		meta+plot word num+title	67.5 (5.0)	63.3 (5.8)	19.6 (5.7)
		meta+title+plot	73.0 (6.1)	67.8 (9.4)	17.8 (6.7)
		meta+plot word num+title+plot	74.2 (2.5)	70.8 (1.8)	14.2 (4.5)
	Concatenation	meta+plot	74.3 (5.5)	70.5 (4.8)	15.6 (4.9)
		meta+plot word num+title	70.0 (7.3)	65.0 (7.2)	20.7 (4.9)
		meta+title+plot	72.3 (9.2)	69.0 (7.8)	14.9 (3.2)
		meta+plot word num+title+plot	74.4 (4.6)	70.6 (3.5)	15.5 (3.9)
	Max Pooling	meta+plot	**74.5 (5.7)**	**70.9 (4.8)**	14.7 (3.8)
		meta+plot word num+title	67.2 (8.8)	62.0 (8.2)	20.4 (4.8)
		meta+title+plot	73.3 (5.3)	68.0 (8.6)	17.2 (7.6)
		meta+plot word num+title+plot	72.3 (7.4)	67.3 (8.3)	18.9 (4.8)

GB のテキストコーパスに基づき事前学習された rinna/japanese-gpt2-medium[15] を採用する．

4. **VHCR-CSGM:** 本研究の提案手法．電子図書館「青空文庫[16]」より作成した約 100 MB のコーパス（14 作家, 2,124 作品, 約 800,000 文）に基づく事前学習済みのモデルを, 表 2 の訓練, 検証データに対して転移学習させることで，データ拡張のための VHCR-CSGM を構築する．転移学習時のモデルへの入力は, 映画の興行的成功予測（'success'）のラベル情報と, 元のあらすじ文の最初と最後の文とする．訓練および検証用データにおけるあらすじ文の平均文数は 25.1 であることを踏まえて, 入力として用いる元のあらすじ文の最初および最後の文と合わせて 25 文となるように, モデルの出力文を 23 文とする．つまり, 生成される新たなあらすじ文は 25 文からなる．訓練時に使用するあらすじ文は, cl-tohoku/bert-base-japanese-whole-word-masking で使用されるトークン化手法に倣い, MeCab+IPAdic+NEologd [59] に基づく MeCab によりトークン化されたのち, WordPiece [60] でサブワードに分割される．また, cl-tohoku/bert-base-japanese-whole-word-masking の 32,000 語の語彙データを使用する．

5. **VHCR:** 従来の VHCR [12] を用いて生成したあらすじ文を用いたデータ拡張手法．モデルへの入力は VHCR-CSGM と異なり，元サンプルのラベル情報と，あらすじ文の最初の文である．そのため，学習時および生成時にの各ステップで最終文の情報は考慮されない．モデルは 24 文を出力し，入力とした元のあらすじ文の最初の文と合わせて，25 文からなる新たなあらすじ文が生成される．なお，VHCR-CSGM と同様に，「青空文庫」より作成したコーパスに基づく事前学習済みのモデルを転移学習させる．

6.1 節で述べた特徴量と分類器の比較実験では，あらすじ文のみ（'plot-only'）を特徴量とした場合には，分類器として LGBM を利用した場合に最も高い予測評価値が得られることを確認した．なお, 分類器を LGBM とした場合の最も高い F1 値は, 特徴量 meta+title+plot を Early Fusion によって結合した場合に得られた．メタ情報と言語情報を両方利用したマルチモーダルな分類においては，特徴量 meta+plot word num+title+plot を Early Fusion によって結合する XGB を用いた場合に最も高い予測評価値が得られた．これらの結果を踏まえて，各テキストデータ拡張手法によって拡張されたあらすじ文を含む訓練データに基づき，以下の分類手法を用いて，表 2 に示したデータセットを対象として映画の興行的成功（'success'）を 2 値分類予測する．

(1) 分類器: XGB，特徴量: plot-only

(2) 分類器: LGBM，特徴量: plot-only

(3) 分類器: XGB，結合手法: Early Fusion，特徴量: meta+title+plot

(4) 分類器: LGBM，結合手法: Early Fusion，特徴量: meta+plot word num+title+plot

6.1 節の実験と同様に，実験は 10 分割交差検証に基づき，試行ごとにモデルのハイパーパラメータを Optuna [57] によって自動調節する．パラメータの値および探索範囲は，表 3 に示したものと同一である．各試行で検証データに対して最も高い評価値（F1 値）を示したモデルを用いてテストデータのラベルを予測する．

6.2.2 評価指標

6.1 節の特徴量と分類器の比較実験と同様に，映画の興行的成功（'success'）の 2 値分類予測における評価指標として Accuracy，F1 値，および MWR を使用する．さらに，生成されたあらすじ文について，（1）元のあらすじ文との語彙的な距離，（2）元のあらすじ文との分散表現の類似度，（3）生成されたあらすじ文の語彙的な多様性に関する，以下の指標を使用して評価する．

BLEU: 生成されたあらすじ文と元のあらすじ文の語彙的な距離を，BLEU [61] によって計測する．BLEU は, 表層類似性に基づく機械翻訳システムの標準的自動評価尺度である．システムが出力する翻訳文集合を $\mathcal{H}$, それに対応する参照文集合を $\mathcal{R}$ とする．システム翻訳文 $h_i \in \mathcal{H}$ には，対応する参照文の集合 $R_i \in \mathcal{R}$ が割り当てられおり，R_i の j 番目の参照文を r_j とする．BLEU は，システムが出力する翻訳文と参照文の N-gram の一致度に基づき，次の式で算出される．

$$\text{BLEU-}N = \text{BP} \cdot \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log P_n\right). \quad (33)$$

BP, P_n は，それぞれ短いシステム翻訳文に対するペナルティと N-gram 適合率であり，以下の

[15] https://huggingface.co/rinna/japanese-gpt2-medium
[16] https://www.aozora.gr.jp/

34, 35 式で算出される．

$$\mathrm{BP} = \begin{cases} 1, & \text{if } \eta > \rho, \\ \exp(1-\rho/\eta), & \text{otherwise}, \end{cases} \tag{34}$$

$$P_n = \frac{\sum_{i=1}\sum_{t_n \in h_i} \min\left(\mathrm{Count}(h_i, t_n), \mathrm{MaxRefCount}(R_i, t_n)\right)}{\sum_{i=1}\sum_{t_n \in h_i} \mathrm{Count}(h_i, t_n)}, \tag{35}$$

ここで，η, ρ は，それぞれシステム翻訳文と参照文の平均文長であり，N は N-gram の語数である．また， $\mathrm{Count}(h_i, t_n)$ はシステム翻訳文 h_i における任意の N-gram t_n の出現頻度，$\mathrm{MaxRefCount}(R_i, t_n)$ は参照文集合 R_i における t_n の出現頻度の最大値 $\max_{r_j \in R_i} \mathrm{Count}(r_j, t_n)$ である．

一般に，BLEU では原文に対して複数の参照翻訳文が存在することを想定している．本研究では，元のあらすじ文を参照文として使用し，各テキストデータ拡張手法により生成されたあらすじ文に対する BLEU を算出する．この BLEU の値が高いほど，生成されたあらすじ文と元のあらすじ文はより語彙的に類似しているといえる．

Shi ら [62] は，BLEU に基づく文生成モデルの評価指標を提案した．まず Forward BLEU（$\mathrm{BLEU_F}$）は，テストデータを参照文として生成文を評価する通常の BLEU と同一である．それに対して Backward BLEU（$\mathrm{BLEU_B}$）は，生成文を参照文として使用し，生成文がテストデータにおける N-gram をどの程度カバーしているかを測定する．直感的には，$\mathrm{BLEU_F}$ は生成モデルの適合率（precision）を測定することを目的とし，$\mathrm{BLEU_B}$ は再現率（recall）を測定することを目的とする違いがある．Harmonic BLEU（$\mathrm{BLEU_{HA}}$）は，36 式で求められ，$\mathrm{BLEU_F}$ と $\mathrm{BLEU_B}$ を組み込んだ総合的な評価指標である．

$$\mathrm{BLEU_{HA}} = \frac{2 \times \mathrm{BLEU_F} \times \mathrm{BLEU_B}}{\mathrm{BLEU_F} + \mathrm{BLEU_B}}. \tag{36}$$

Plot Vector Similarity（PVS）： 生成されたあらすじ文と元のあらすじ文に対するそれぞれの分散表現の距離を Plot Vector Similarity（PVS）により計測する．具体的には，4.1 節で示した手法により取得した，元のあらすじ文とテキストデータ拡張手法により生成されたあらすじ文に対するそれぞれの分散表現の余弦類似度を，以下の 37 式で算出する．

$$\mathrm{PVS}(v_{\mathrm{org}}, v_{\mathrm{aug}}) = \frac{v_{\mathrm{org}}^{\top} v_{\mathrm{aug}}}{\| v_{\mathrm{org}} \| \quad \| v_{\mathrm{aug}} \|}, \tag{37}$$

ここで，v_{org}, v_{aug} はそれぞれ元のあらすじ文および生成されたあらすじ文に対する分散表現である．BERT に基づき取得された分散表現には，あらすじ文の意味的な情報が圧縮されていると考えられるため，PVS の値が高いほど，元のあらすじ文と生成されたあらすじ文はより意味的に類似しているといえる．

Distinct-N（DIST-N）： 生成されたあらすじ文の語彙的な多様性に関して，Distinct-N（DIST-N）[63] の値により評価する．DIST-N は，以下の式で算出される．

$$\text{DIST-}N(G) = \frac{\mathrm{Unique}_N(G)}{\mathrm{Count}_N(G)}, \tag{38}$$

ここで $G = \{T_1, T_2, \ldots, T_K\}$ は K 個のあらすじ文の集合を表し，Unique_N はユニークな N-gram の数，Count_N はすべての N-gram の数を表す．DIST-N の値が高いほど，より語彙的に多様なあらすじ文が生成されたといえる．

6.2.3 結果と考察

映画の興行的成功予測： 表 6 に，各手法により拡張した訓練データに基づくモデルのテスト予測評価値を示す．拡張したあらすじ文に対する分散表現のみ（‘plot-only’）を特徴量としたユニモーダル分類の結果について，分類器を XGB および LGBM としたいずれの場合も，EDA を用いて拡張されたあらすじ文に基づいて訓練されたモデルに対する予測評価値が最も高くなった．また，提案手法である VHCR-CSGM を用いた場合についても，データ拡張前と比較して高い予測評価値が得られたことから，手法の有効性は確認できた．

さらに，拡張したあらすじ文と元のサンプルのメタ情報およびタイトルを特徴量としたマルチモーダル分類の結果について，分類器を XGB および LGBM としたいずれの場合も，提案手法である VHCR-CSGM を用いた場合の Accuracy および F1 値がそれぞれ最大となった．この結果は，映画の興行的成功予測において，特徴量であるあらすじ文の拡張手法として提案手法が高い性能を有することを示唆している．また，特に LGBM を分類器とした場合について，すべての手法によって拡張された訓練データに基づくモデルの予測性能が大きく向上した．データ拡張前のあらすじ文のみを特徴量とする予測実験において，分類器

表 6: 各テキストデータ拡張手法により拡張した訓練データに基づくモデルのテスト予測評価値

分類器	結合手法	特徴量	拡張手法	Accuracy (↑)	F1 値 (↑)	MWR (↓)
XGB	Unimodal	plot-only	拡張なし	70.2 (2.5)	66.4 (2.2)	17.1 (2.4)
			EDA	**70.7 (2.1)**	**68.1 (2.2)**	**14.9 (1.9)**
			CBERT	70.2 (1.4)	67.3 (1.9)	16.1 (3.2)
			GPT-2_{context}	68.9 (2.4)	65.8 (3.7)	16.5 (3.7)
			VHCR	69.6 (2.2)	66.7 (3.0)	15.9 (3.6)
			VHCR-CSGM（提案手法）	70.6 (1.8)	67.3 (1.8)	16.3 (2.1)
	Early Fusion	meta+plot word num+title+plot	拡張なし	75.5 (2.4)	71.4 (2.2)	15.9 (1.8)
			EDA	75.4 (2.2)	71.6 (2.3)	15.6 (2.9)
			CBERT	75.9 (2.5)	72.5 (2.7)	13.6 (3.1)
			GPT-2_{context}	74.9 (2.8)	72.0 (2.7)	**12.8 (2.5)**
			VHCR	75.4 (1.2)	71.5 (1.9)	15.1 (4.6)
			VHCR-CSGM（提案手法）	**76.4 (2.7)**	**73.3 (3.6)**	12.9 (4.9)
LGBM	Unimodal	plot-only	拡張なし	71.8 (1.6)	68.1 (2.3)	15.8 (2.8)
			EDA	**74.2 (1.3)**	**71.5 (1.8)**	**13.3 (1.5)**
			CBERT	72.5 (2.0)	69.7 (2.7)	13.7 (3.4)
			GPT-2_{context}	71.2 (1.1)	68.2 (1.6)	15.3 (2.8)
			VHCR	71.7 (1.7)	68.6 (2.2)	15.4 (1.1)
			VHCR-CSGM（提案手法）	72.7 (2.4)	69.7 (2.9)	14.2 (2.2)
	Early Fusion	meta+title+plot	拡張なし	73.1 (1.4)	70.6 (1.6)	12.8 (3.3)
			EDA	75.1 (1.3)	72.6 (1.7)	11.9 (1.7)
			CBERT	76.4 (1.1)	73.9 (1.7)	11.3 (3.3)
			GPT-2_{context}	76.2 (2.4)	72.7 (4.2)	12.2 (3.8)
			VHCR	76.5 (1.4)	73.8 (1.7)	11.2 (1.1)
			VHCR-CSGM（提案手法）	**77.3 (1.0)**	**74.7 (1.1)**	**10.3 (1.2)**

として LGBM を用いた場合に最大の評価値が得られたことから，他の分類器を用いる場合よりもあらすじ文の言語情報が重要になることが推察される．

生成されたあらすじ文の評価: 表 7 に，元のあらすじ文を参照文とする，各テキストデータ拡張手法により生成したあらすじ文に対する BLEU-N および PVS の値を示す．EDA に対する BLEU，PVS の値がそれぞれ最大となり，それに次いで CBERT に対する BLEU および PVS の値が高くなった．EDA と CBERT は，元のあらすじ文における単語を置換する操作に基づく手法であるため，元のあらすじ文との語彙的，意味的な類似度が高くなるのは妥当といえる．

生成的なテキストデータ拡張手法である GPT-2_{context}，VHCR，および VHCR-CSGM を比較すると，VHCR-CSGM に対する BLEU，PVS の値が最大であり，元のあらすじ文と語彙的および意味的に最も類似したあらすじ文を生成することが確認された．また，GPT-2_{context} に対する BLEU，PVS の値が最小となった．

図 14-18 に，各テキストデータ拡張手法を用いて拡張したあらすじ文に対する 768 次元の分散表現を，t 分布型確率的近傍埋め込み法（t-Distributed Stochastic Neighbor Embedding, t-SNE）[64] により 2 次元に圧縮した結果を示す．図 14 および図 15 より，置換的拡張手法である EDA および CBERT によって生成されたあらすじ文に対する分散表現（正例: ▲，負例: ▲）は，元のあらすじ文の分散表現（正例: ●，負例: ●）とほとんど重複して描画されている．一方で，図 16 に示した GPT-2_{context} を用いて生成したあらすじ文の分散表現は，元のあらすじ文の分散表現と離れた位置にクラスタを形成していることが確認できる．さらに図 17 に示した，VHCR により生成されたあらすじ文の分散表現についても，元のあらすじ文

表 7: 生成されたあらすじ文の評価

評価指標	EDA	CBERT	GPT-2_{context}	VHCR	VHCR-CSGM（提案手法）
BLEU_{F}-2	**0.847**	0.432	0.0501	0.248	0.314
BLEU_{F}-3	**0.791**	0.420	0.0342	0.178	0.251
BLEU_{F}-4	**0.741**	0.408	0.0259	0.139	0.212
BLEU_{F}-5	**0.697**	0.397	0.0211	0.116	0.186
BLEU_{B}-2	**0.847**	0.524	0.121	0.260	0.331
BLEU_{B}-3	**0.790**	0.509	0.0824	0.187	0.264
BLEU_{B}-4	**0.741**	0.495	0.0623	0.146	0.223
BLEU_{B}-5	**0.697**	0.482	0.0506	0.121	0.196
BLEU_{HA}-2	**0.847**	0.474	0.0708	0.253	0.322
BLEU_{HA}-3	**0.791**	0.460	0.0484	0.182	0.257
BLEU_{HA}-4	**0.741**	0.447	0.0366	0.142	0.217
BLEU_{HA}-5	**0.697**	0.435	0.0298	0.118	0.191
PVS	**0.9805**	0.9782	0.9018	0.9146	0.9154
DIST-2	0.148	0.0857	**0.197**	0.0672	0.0680
DIST-3	0.316	0.170	**0.497**	0.182	0.188
DIST-4	0.425	0.220	**0.718**	0.314	0.325

の分散表現とは離れた位置に描画された．また図 18 は，提案手法である VHCR-CSGM を用いて生成したあらすじ文の分散表現を表している．生成的拡張手法である GPT-2_{context}，VHCR，および VHCR-CSGM を比較すると，VHCR-CSGM を用いて生成したあらすじ文は，元のあらすじ文の分散表現と最も接近して描画されていることが観察できる．

生成されたあらすじ文の語彙的な多様性: 表 7 に，元のあらすじ文と生成されたあらすじ文の分散表現に対する PVS の値および生成されたあらすじ文に対する DIST-N の値を示す．GPT-2_{context} により生成されたあらすじ文に対する DIST-N の値がそれぞれ最大となった．GPT-2_{context} に次いで高い値を得た EDA は，あらすじ文中のトークンに対して無作為に変更を加えるため，文章のコンテキストを保持する必要がなく，より多様な N-gram をもつあらすじ文を生成すると考えられる．

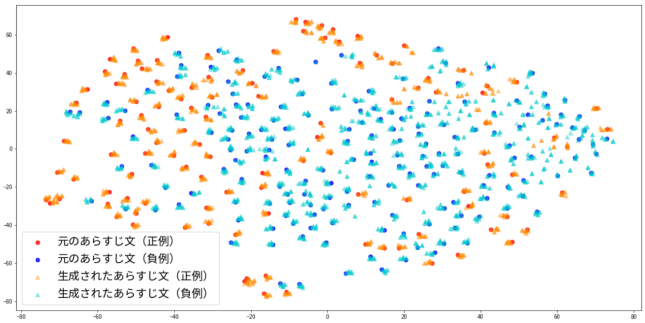


図 14: EDA を用いて拡張したあらすじ文に対する分散表現

7 結言

本研究では，まず 2 章で映画の興行的成功予測の研究分野の現状を示し，以下のように課題を整理した．

(1) 日本語のあらすじ文を含んだ映画の興行収入に関するデータセットが整備されていない．

(2) 映画のメタ情報のみ，あるいはあらすじ文の言語情報のみを特徴量として使用する研究は存在しているが，その両方を使用するマルチモーダルな予測手法はほとんど検討されていない．

上記の課題に対して，日本語のあらすじ文を含む映画の興行的成功予測のためのデータセット新たに構築した．3 章では，構築したデータセットの特徴量について述べた．続いて 4 章では，提案手法である，映画のメタ情報と，BERT を用いて処理したあらすじ文の言語情報の両方を用いたマルチモーダルな映画の興行的成功

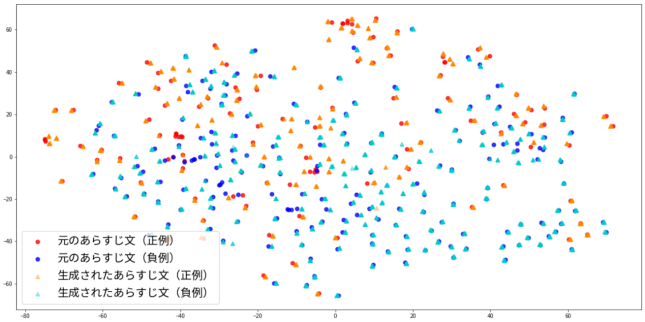


図 15: CBERT を用いて拡張したあらすじ文に対する分散表現

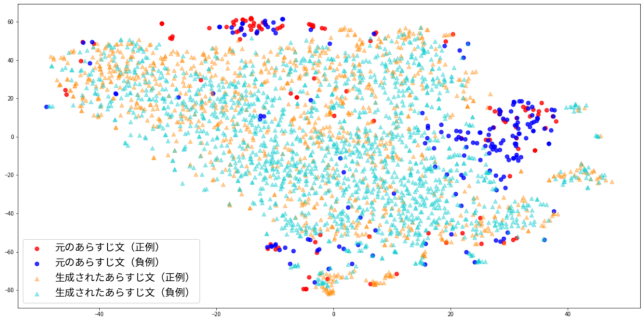


図 17: VHCR を用いて拡張したあらすじ文に対する分散表現

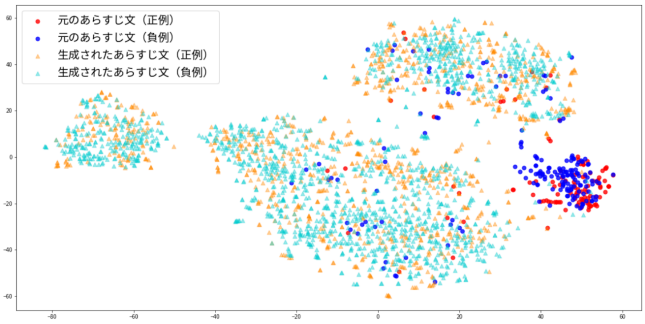


図 16: GPT-2_{context} を用いて拡張したあらすじ文に対する分散表現

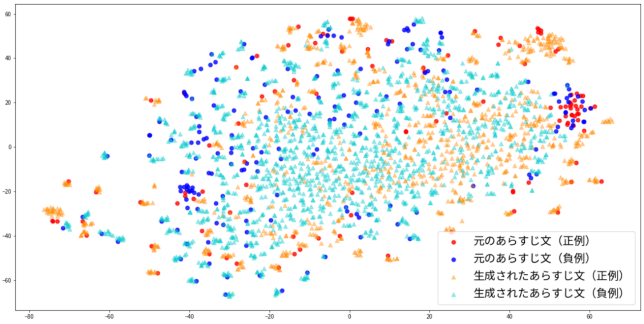


図 18: VHCR-CSGM を用いて拡張したあらすじ文に対する分散表現

予想モデルについて詳述した．また，構築したデータセットのサイズが予測モデルの性能向上のボトルネックになると考え，テキストデータ拡張手法を用いてあらすじ文データを拡張した．5 章では，技術的背景と合わせて，提案手法である VHCR-CSGM と，それに基づくあらすじ文拡張について述べた．

そして 6 章では，映画の興行的成功予測モデルと，VHCR-CSGM に基づくあらすじ文データ拡張の有効性を確認する数値実験について述べた．実験の結果は，映画のメタ情報だけでなく，あらすじ文およびタイトル名の言語情報を特徴量として用いることで予測精度が向上することを示唆していた．さらに，従来の置換的拡張手法および生成的拡張手法を用いた場合よりも，VHCR-CSGM を用いて拡張したあらすじ文データを含む訓練データに基づく予測モデルは，高い性能を有することを実験的に確認した．

本研究では，映画のあらすじ文データのみを拡張し，元のサンプルと同一のメタ情報を使用して新たなサンプルを生成した．そのため，より多様かつ妥当なサンプルを生成するために，映画のメタ情報を考慮したデータ拡張に改良の余地があるといえる．例えば，メタ情報を擬似的に生成する操作や，メタ情報に基づいてあらすじ文を生成する操作の検討が，今後の課題として挙げられる．

参考文献

[1] Michael T. Lash and Kang Zhao. Early predictions of movie success: The who, what, and when of profitability. *Journal of Management Information Systems*, Vol. 33, pp. 874–903, 2016.

[2] Travis Ginmu Rhee and Farhana Zulkernine. Predicting movie box office profitability: A neural network approach. In *Proceedings of the 15th IEEE International Conference on Machine Learning and Applications*, pp. 665–670, 2016.

[3] Marta Galvão and Roberto Henriques. Forecasting movie box office profitability. *Journal of Information Systems Engineering and Management*, Vol. 3, , 2018.

[4] Quazi Ishtiaque Mahmud, N. Z. Shuchi, Fazle Mohammed Tawsif, Asif Mohaimen, and Ayesha Tasnim. A machine learning approach to predict movie revenue based on pre-released movie metadata. *Journal of Computer Science*, Vol. 16, pp. 749–767, 2020.

[5] You Jin Kim, Yun Gyung Cheong, and Jung Hoon Lee. Prediction of a movie's success from plot summaries using deep learning models. In *Proceedings of the 2nd Workshop on Storytelling (ACL-2019)*, pp. 127–135, 2019.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[7] David Bamman, Brendan O'Connor, and Noah A. Smith. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 352–361, 2013.

[8] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 6382–6388, 2019.

[9] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional bert contextual augmentation. In *International Conference on Computational Science*, pp. 84–95, 2019.

[10] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 7383–7390, 2020.

[11] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems (ACL-2020)*, pp. 18–26, 2020.

[12] Yookoon Park, Jaemin Cho, and Gunhee Kim. A hierarchical latent structure for variational conversation modeling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1792–1801, 2018.

[13] 飯倉陸, 岡田真, 森直樹. 潜在変数付き階層型エンコーダ・デコーダモデルに基づく物語の補完的生成手法の提案. 人工知能学会全国大会論文集, 3D4OS12c03, 2021.

[14] Riku Iikura, Makoto Okada, and Naoki Mori. Cvae-based complementary story generation considering the beginning and ending. In *Distributed Computing and Artificial Intelligence, Volume 1: 18th International Conference*, pp. 22–31, 2022.

[15] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 3776–3783, 2016.

[16] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 3295–3301, 2017.

[17] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2014.

[18] Ting Liu, Xiao Ding, Yiheng Chen, Haochen Chen, and Maosheng Guo. Predicting movie box-office revenues by exploiting large-scale social media content. *Multimedia Tools and Applications*, Vol. 75, pp. 1509–1528, 2014.

[19] Taegu Kim, Jungsik Hong, and Pilsung Kang. Box office forecasting using machine learning algorithms based on SNS data. *International Journal of Forecasting*, Vol. 31, No. 2, pp. 364–390, 2015.

[20] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, 2018.

[21] C. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 8, pp. 216–225, 2014.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[23] Rafael PÉrez Ý PÉrez and Mike Sharples. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 13, No. 2, pp. 119–139, 2001.

[24] Hugo Liu and Push Singh. MAKEBELIEVE: using commonsense knowledge to generate stories. In *Proceedings of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence*, pp. 957–958, 2002.

[25] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, 2016.

[26] Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. LSDSem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (ACL-2017)*, pp. 46–51, 2017.

[27] Melissa Roemmele, Sosuke Kobayashi, Naoya Inoue, and Andrew Gordon. An RNN-based binary classifier for the story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (ACL-2017)*, pp. 74–80, 2017.

[28] Jian Guan, Yansen Wang, and Minlie Huang. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 6473–6480, 2019.

[29] Prakhar Gupta, Vinayshekhar Bannihatti Kumar, Mukul Bhutani, and Alan W Black. WriterForcing: Generating more interesting story endings. In *Proceedings of the Second Workshop on Storytelling (ACL-2019)*, pp. 117–126, 2019.

[30] Yusuke Mori, Hiroaki Yamane, Yusuke Mukuta, and Tatsuya Harada. Finding and generating a missing part for story completion. In *Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pp. 156–166, 2020.

[31] Parag Jain, Priyanka Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha, and Karthik Sankaranarayanan. Story generation from sequence of independent short descriptions. In *arXiv preprint arXiv:1707.05501*, 2017.

[32] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, 2018.

[33] Lara Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark Riedl. Event representations for automated story generation with deep neural nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, 2018.

[34] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 7378–7385, 2019.

[35] Faeze Brahman and Snigdha Chaturvedi. Modeling protagonist emotions for emotion-aware storytelling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 5277–5294, 2020.

[36] Tri Dao, Albert Gu, Alexander J. Ratner, Virginia Smith, Christopher De Sa, and Christopher Ré. A kernel theory of modern data augmentation. In *arXiv preprint arXiv:1803.06084*, 2019.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, Vol. 25, 2012.

[38] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Vol. 27, 2014.

[39] Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 198–202, 2017.

[40] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

[41] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[42] 工藤拓, 山本薫, 松本裕治. Conditional random fields を用いた日本語形態素解析. 情報処理学会研究報告自然言語処理（NL）, Vol. 161, pp. 89–96, 2004.

[43] 松田寛, 大村舞, 浅原正幸. 短単位品詞の用法曖昧性解決と依存関係ラベリングの同時学習. 言語処理学会第 25 回年次大会発表論文集, pp. 201–204, 2019.

[44] 松田寛. Ginza - universal dependencies による実用的日本語解析. 自然言語処理, Vol. 27, No. 3, pp. 695–701, 2020.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998–6008. 2017.

[46] Raghavendra Pappagari, Piotr Żelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. Hierarchical transformers for long document classification. In *arXiv preprint arXiv:1910.10781*, 2019.

[47] Leo Breiman. Random forests. *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.

[48] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[49] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, Vol. 30, pp. 3146–3154, 2017.

[50] Chi Thang Duong, Remi Lebret, and Karl Aberer. Multimodal classification for analysing social media. In *arXiv preprint arXiv:1708.02099*, 2017.

[51] Ferda Ofli, Firoj Alam, and Muhammad Imran. Analysis of social media data using multimodal deep learning for disaster response. In *17th International Conference on Information Systems for Crisis Response and Management*, 2020.

[52] Tomas Mikolov, MartinKarafiát, LukásBurget, JanCernocký, Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pp. 1045–1048, 2010.

[53] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua 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*, pp. 1724–1734, 2014.

[54] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese Computational Linguistics*, pp. 194–206, 2019.

[55] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, 2011.

[56] Canhui Wang, Min Zhang, Shaoping Ma, and Liyun Ru. Automatic online news issue construction in web environment. In *Proceedings of the 17th International Conference on World Wide Web*, pp. 457–466. Association for Computing Machinery, 2008.

[57] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[58] Hitoshi Isahara, Fransis Bond, Kiyotaka Uchimoto, Masao Utiyama, and Kyoko Kanzaki. Development of the japanese wordnet. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, 2008.

[59] Toshinori Sato, Taiichi Hashimoto, and Manabu Okumura. Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval (in japanese). In *Proceedings of the 23rd Annual Meeting of the Association for Natural Language Processing*, pp. NLP2017–B6–1, 2017.

[60] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv preprint arXiv:1609.08144*, 2016.

[61] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[62] Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. Toward diverse text generation with inverse reinforcement learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4361–4367. International Joint Conferences on Artificial Intelligence Organization, 2018.

[63] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, 2016.

[64] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, Vol. 9, No. 86, pp. 2579–2605, 2008.