2022年度博士学位論文

# マルチモーダル情報の対応関係を捉えたモダリティ変換

お茶の水女子大学大学院

人間文化創成科学研究科

理学専攻

濵園侑美

2023年　3月

# 要旨

人は自然言語を介して物事をより深く理解する．近年，計算機の発達により，言葉や画像，数値情報などを扱う手法が発達し，その能力が向上している．さまざまな物事を計算機上で扱えるようになった一方，それぞれの物事はモダリティ毎に独立して扱われていたため，モダリティ間の関係性を捉えるに至っていない．そこで，本論文は言葉を中核に据えて，異なるモダリティの対応関係を捉え，生成する課題に取り組む．

一つ目は言語から非言語系列を生成する課題として，自然言語からロボット動作を生成する課題に取り組む．言語を計算機上で扱う手法として，意味の捉え方に文章中の単語の頻度情報や周辺単語による単語の類似度により自動的に意味を付与する分散意味表現を用いる．非言語系列として，本課題ではロボット動作を扱う．まず，人間の関節の動きとは異なるロボットに動作を生成させるためにロボットが行える基本動作を組み合わせることにより，人の動きを真似て行える枠組みを提案する．また，それらの基本動作を組み合わせることにより，複雑な動作を表現，生成することを可能とする．これらを達成したのち，言葉と動作の対応関係を学習する課題に取り組む．2つの対応関係を捉える方法として，本課題ではニューラルネットワークを用いる．特に，多様なロボット動作と曖昧な表現との対応関係を学習できるようなニューラルネットワークの構造を提案し，検証する．

二つ目は非言語から言語を生成する課題として，一般ドメインの動画の実況生成課題に取り組む．本課題に則したデータセットは存在しないため，既存のビデオコーパスを用いて実況の収集を行い，データセットの構築を行う．その後，課題をタイミング推定と発話生成の 2 つのサブタスクに分割して，動画のみから実況を生成する課題の解決に取り組む．これは，実況は映像に合わせて出力されるテキストのため，実況を開始するタイミングや，実況テキストの長さを制御する必要があるためである．タイミング推定には，人の行動を推定できる枠組みを用いる．発話生成には，言語生成に適したニューラルネットワークモデル，特に Transformer を用いる．

最後に，非言語から言語を生成する課題のうち，実世界で得られたデータとテキストを用いる場合に発生する問題について取り上げ，問題の原因を分析し，解決策を提案する．特に，時系列数値データの概況テキスト生成における問題点を取り上げる．既存の時系列数値データの概況テキスト生成に用いるデータセットには，時系列データの data-to-text に特有の参照時刻の不整合問題と，実世界のデータとテキストを用いた場合に起こりうる，入力データから予測できない属性を含む出力を求められる問題が含まれている．データセッ

トの作成手順や入出力データの特性を理解することで，既存の枠組みを大きく変えることなく，これらの問題を解決する方法を提案する．

以上三つの取り組みにより，言葉を中心としてモダリティーの方向性を変更する手法の提案と，方向性を示す．

キーワード：自然言語処理，機械学習，ヒューマノイドロボット，ニューラルネットワーク，言語生成，**data-to-text**，**vision-to-text**

# Abstract

People understand things more deeply through natural language. In recent years, the development of computers has led to the development of methods for handling natural language, images and numerical information, and their capabilities have improved. While various things can now be handled on computers, each thing has been handled independently for each modality, and the relationship between modalities has not yet been captured. Therefore, this paper addresses the issue of capturing and generating correspondence between different modalities, with centered on language.

The first task is generating non-verbal sequences from language, the task of generating robot behaviours from natural language. As a method for handling language on a computer, the meaning of sentences are captured by using the frequency information of words in the sentence and the similarity of words in the surrounding words to create a distributed semantic representation that automatically assigns a meaning. Distributed representation is used to automatically assign meanings based on the frequency information of words in a sentence and the similarity of words by surrounding words. As a non-verbal series, this task deals with robot movements. First, in order to generate movements for the robot that are different from the movements of human joints ,we propose a framework that can imitate human movements by combining basic movements that can be performed by a robot. By combining these basic movements, it is possible to express and generate complex movements. After achieving these goals, the task of learning the correspondence between words and actions is tackled. As a method for capturing the correspondence between the two, we use neural networks in this task. In particular, we propose and validate a neural network structure that can learn the correspondence between various robot actions and ambiguous expressions.

Secondly, as a task to generate language from non-verbal language, we tackle the task of generating live video in the general domain. As there is no dataset available for this task, the dataset is constructed by collecting the actual situation using an existing video corpus. The task is then divided into two subtasks, timing estimation and speech generation, to solve the task of generating the actual situation from the video alone. This is because the actual situation is text that is output along with the video, so it is necessary to control the timing of the start of the actual situation and the length of the actual situation text. For

timing estimation, a framework that can estimate human behaviour is used. For speech generation, we use neural network models suitable for language generation, in particular Transformer.

Finally, we address the problem of generating language from non-verbal data, which arises when using real-world data and text, analyse the causes of the problem and propose a solution. In particular, we address the problem of generating summary text from time-series numerical data. Existing datasets for generating summary text for time-series numerical data include The problem of inconsistent reference times inherent to data-to-text for time-series data, and The problem of being asked to produce output containing attributes that cannot be predicted from the input data, which can occur when using real-world data and text. By understanding the procedures for creating datasets and the characteristics of input and output data, we propose a way to solve these problems without significantly changing the existing framework.

Through these three efforts, we propose a method for changing the direction of modality with a focus on natural language, and discuss the future direction of the method.

**Keywords: natural language， machine learning， humanoid robot， neural network， natural language generation， data-to-text， vision-to-text**

# 目 次

# 図 目 次

# 表 目 次

# 第1章　序論

## 1.1　背景

人工知能の研究とは，これまで人間にしかできなかった知的行為を機械的に実行するにはどうしたら良いか，人間のような知を構成するにはどうすれば良いかを探究する学問である[1]．つまり究極的には，人工知能の研究は人間の外界認識方法や外界との関係性の持ち方を機械で実行できるようにすることである．人間は物を見たり，匂いを嗅いだり，音を聞いたりと，外界を複数種類の信号で捉えている．機械が人間を取り巻く外界を捉えるためには，このような信号をまとめて解釈できるようにする必要がある．

様々な数値，画像，音声，言語などのデータ種別のことをモダリティ（*Modality*）と呼び，異なるモダリティとの関係性を捉えたり，複数のモダリティを組み合わせて用いる課題をマルチモーダル（*Multimodal*）情報を用いた課題と呼ぶ．近年，計算機の発達，特に GPU の登場により，ニューラルネットワークを用いた各モダリティでの課題解決手法の発展が目覚ましい．例えば，2010 年から始まった大規模画像認識のコンペティション ImageNet Large Scale Visual Recognition Challenge(ILSVRC）において，Krizhevsky et al. (2012) は，深層学習を用いた AlexNet により従来の学習方法による画像認識精度を上回り，さらに二位を大きく引き離して圧勝した．また Mnih et al. (2013) は，強化学習の行動価値関数 Q の学習に深層学習を用いた Deep Q-Network を提案し，Arcade Learning Environment (ALE) の 7 つの Atari 2600 games のうち，6 つのゲームで従来の学習手法のスコアを上回り，さらに 3 つのゲームで人間を上回った．計算機上で言語を扱う方法もニューラルネットワークを用いた手法が登場し，特に word2vec (Mikolov et al., 2013a) が発表されて以来，その技術は急激に発展している．

各モダリティで独立して発展したニューラルネットワークによる課題解決手法を，複数のモダリティを処理して関連付けるマルチモーダルの課題に適用しようと，現在さまざまなマルチモーダル課題の精度を競う大会の開催やデータセットの公開が行われている．例えば，2016 年より開催されている International Challenge on Activity Recognition[2]は映像と人の動作の関係性を問うマルチモーダルな課題を設定しており，人間の動作を集めた動画コーパス ActivityNet の動画から動作認識を行う．この動作認識精度を競う大会のベースラ

[1]近年では人間を超える精度を持つ人工知能も出現しており，必ずしも「人間の」知を構造すること目的ではない場合もある (松原, 2020)

[2]http://activity-net.org/challenges/2016/index.html

インモデルでは，映像の特徴量抽出にニューラルネットワークの C3D (Tran et al., 2015) を用いている．

マルチモーダル情報を用いた課題には，次の 5 つの技術的課題 (Baltrušaitis et al., 2019) が存在し，マルチモーダル課題には，これらの技術的課題が組み合わさっている．

1. **表現**（Representation）
   マルチモーダルデータの表現方法や要約方法についての課題．異なるモダリティのデータを同一空間で扱えるか，なども課題の一つ．

2. **変換**（Translation）
   あるモダリティから別のモダリティにデータを変換（マッピング）する課題．画像説明文生成など正解が一つに絞られない場合も存在する．

3. **直接的な関係性獲得**　（Alignment）
   複数のモダリティ間の直接的な関係性を明らかにする課題．例えば，レシピ手順と調理動作の一致を獲得する課題などが含まれる．

4. **融合**（Fusion）
   ある予測行うために複数のモダリティのデータを用いる課題．例えば，唇の動きを含む動画とその音声を用いた音声認識課題などが含まれる．

5. **共学習**（Co-learning）
   あるモダリティ内で作成された予測モデルや表現方法を別のモダリティに転移する課題．共学習，概念的接地，Zero-shot learning などの手法を用いた課題．

マルチモーダル情報を用いた課題において，特に 2. 変換は音声合成 (Hunt and Black, 1996) や動画説明文生成 (Kojima et al., 2002) などが長年行われている．近年は，各モダリティでのニューラルネットワークによる課題解決手法の発展と，画像やテキストを組み合わせたデータセットである COCO dataset (Chen et al., 2015) に代表されるマルチモーダルな大規模データセットの登場により，益々発展している．マルチモーダルな課題における変換課題には，以下の 2 つのアプローチ方法がある．

- **Example-based approaches**
  あらかじめ用意した辞書を元に，近いサンプルを検索する手法

- **Generative approaches**
  入力データを元に別モダリティのデータを生成する手法

Generative approaches には，入出力データの両方のモダリティの性質を理解したうえで，出力データを生成する必要があり，構造がより複雑になることから，初期のマルチモーダル情報のモダリティ変換課題では，Example-based approaches が採用されていた．しかし，

Example-based approaches は辞書全体がモデルであるため，モデルが大きくなるほど推論が遅くなり，また課題が単純であるか辞書が非常に大きくない限り，入力データに対応する正確なサンプルが辞書に常に存在するとは限らないという問題がある．最近では精密な画像の生成が可能なモデル (Goodfellow et al., 2014) や，より流暢な文章の生成が可能なモデル (Bahdanau et al., 2015) の登場など，ニューラルネットワークによる各モダリティでの生成手法の発展により，Example-based approaches における問題を解決できる Generative approaches が採用されつつある．

Generative approaches によるマルチモーダル情報の変換手法は，まず入出力データをそれぞれのモダリティの性質を反映した表現方法を用いて表現した上で，入出力の関係性を捉えたモデルと出力データに適した出力手法によって生成を行うため，限られたモダリティ間の変換に特化した手法となっている．そのため，既存のモダリティ変換手法を異なるモダリティにはそのまま適用できない．さらに，Generative approaches で特にニューラルネットワークを用いる場合には，大規模なデータセットが必要となる．大規模データセットは画像とその説明文データセット (Young et al., 2014; Chen et al., 2015) や動画と人の動作ラベルのデータセット (Fabian Caba Heilbron and Niebles, 2015)，テキストと読み上げ音声のデータセット (Panayotov et al., 2015; Köhn et al., 2016; Ardila et al., 2020) が知られており，それぞれ画像説明文生成課題や動作認識課題，音声認識課題に用いることができる．これら大規模データセットが公開されているマルチモーダル課題ではモダリティ変換手法の研究が進められている一方，それ以外のモダリティでは変換手法が十分に明らかになっていない．また，データセットが固有のドメインに特化している場合には，モダリティ変換は限られたドメインに特化した手法となる場合もある (Tanaka-Ishii et al., 1998; Ishigaki et al., 2021)．異なるドメインや新たなモダリティ間の変換に適切なモデルを得るには，新たにデータセットを作成した上で，入出力データの適切な表現方法を適用し，モデルを構築する必要がある．

また，Generative approaches によるマルチモーダル情報の変換には，音声認識のように入力に対応する正解が 1 つに限られる課題がある一方，音声合成や画像説明文生成など，複数の正解が存在しうる課題もある．このような正解が一つに決まらない課題には，人手評価を行う (van den Oord et al., 2016; Kulkarni et al., 2013; Venugopalan et al., 2015b) などの評価方法の工夫や，モダリティ変換の方向性を設定するなどの課題の再設定を行う必要がある．

## 1.2 本研究の目的

本研究ではマルチモーダル情報を用いた課題のうち，ニューラルネットワークを用いた Generative approaches による変換課題に取り組む．特にモダリティ変換を実用に向けて行う．現在，計算機が発達したことで，大量の情報が得られるようになった一方，人手不足により得られた情報を適切に処理する人員が足りず，的確に利用できていない．そのため，

分析を人手を借りず，機械的に行う必要性が増している．モダリティ変換を実用に向けて行うことは，このような人手不足の問題への解決の手掛かりとなりうる．

また人間は言語によって，物事をより深く理解する傾向にあるため，知識の共有や保存を自然言語を通じて行うことは非常に重要である．言語・非言語間のモダリティ変換が十分に達成されれば，今まで個人が個々に持つ知識や勘などの優れたナレッジを自然言語で共有して活用できるようになり，さらなる技術の向上や，業務効率化，生産性の向上が可能となる．そのため，本研究ではマルチモーダルのモダリティ変換のうち，言語を中心に据えた変換課題に取り組み，言語と非言語のモダリティ変換の適用手法を明らかにする．

さらに，言語から非言語，非言語から言語へのモダリティ変換をそれぞれ扱い，また非言語データを複数扱うことで，言語と特定のモダリティ間における言葉の曖昧さや言葉の広がりを許容するモダリティ変換モデルの獲得を目指す．

## 1.3 課題設定

はじめに，言語から非言語を生成する課題として，自然言語からロボット動作を生成する課題に取り組む．これは，１つの言葉に対して動作が複数あっても許容する，言葉の曖昧さをモダリティ変換によって扱う課題である．自然言語からロボット動作を生成する課題は 3 つのサブタスクに分けられる．一つ目は，言葉の意味を捉えるというサブタスクである．言葉の意味を計算機上で扱うため，自然言語処理の技術を用いる．文章の意味を捉える方法は構文解析等の技術が知られている．一方，言葉の意味を捉える方法は，単語が元々持っている意味や意味的関係から意味ネットワークを生成する等の方法がある．しかし，それらは全て人手で作成した辞書がなくてはならない．そこで本課題では，意味の捉え方に文章中の単語の頻度情報や周辺単語による単語の類似度を用い，自動的に意味を付与する分散意味表現を用いる．それにより，曖昧な言語表現であっても分散意味表現上では一意に定まり，扱いが容易になる．二つ目に，ロボットに動作を生成させるというサブタスクである．ロボットを用いる際に留意しなければならないこととして，ロボットの関節の動きと人間の関節の動きは異なるということである．そこでロボットが行える基本動作を組み合わせることにより，人の動きを真似て行える枠組みを提案する．また，それらの基本動作を組み合わせることにより，複雑な動作を表現，生成することを可能とする．最後に，言葉と動作の対応関係を学習するサブタスクである．2 つの対応関係を捉える方法は様々あるが，本課題ではニューラルネットワークの技術を用いる．また，多様なロボット動作と曖昧な表現との対応関係を学習する枠組みを提案し検証する．以上の 3 つのサブタスクを達成することで，人の言葉による指示からロボットが動作を生成し動作することが可能となる．また，初めて行う動作であっても言葉の意味から推測し，ロボットが動作生成することが可能となる．

次に非言語から言語を生成する課題として，動画像データから自然言語文を生成する課

題，特に一般ドメインの動画に対する実況生成課題に取り組む．ドメインを限らないオープンドメインの課題とすることで，ある事象に対して言葉の広がりを許容する課題となる．動画や画像に対して自然言語でその内容を記述する研究は現在広く行われており，本課題で取り組む動画の実況生成もその一つである．主な動画からの自然言語生成の課題は，動画内のイベントを抽出し，そのイベントに対して説明文（キャプション）を付与する課題が挙げられる．一方，本課題の対象である実況は，視聴者に映像の内容をわかりやすく伝えるために，映像と共に音声として出力できたり，映像に被せてテキストが提示されたりするテキストである．そのため実況生成では，実況を開始するタイミングや，実況の長さを制御する必要がある．このような観点から，実況生成を扱った研究では，タイミング推定と発話生成の２つのサブタスクに分けて課題解決している．実況生成に関する既存研究は，実況の特性上，例えばスポーツの試合の実況など，特定の分野を対象とするものに限られていた．このような特定の分野を対象として実況生成を行う課題の場合，分野特有の情報を用いて実況生成を行う場合がほとんどである．しかし，本課題では分野特有の情報を用いることができず，映像とテキストのみを用いるため，より難しい課題設定と言える．

最後に，非言語から言語を生成する課題のうち，実世界で得られたデータとテキストを用いる場合に発生する問題について取り上げ，問題の原因を分析し，解決策を探る．限られたデータから適切に目的のテキストを生成する手法の獲得であり，複数の解釈が可能なデータから，言葉によって解釈の方向性を決定する課題とも言える．時系列数値データを用いてテキスト生成を行う課題では，アライメントが不十分な場合，データとテキストに不整合が生じ，それにより生成文で誤りが発生することがあるとされている．また，実世界で得られたデータとテキストを使ってデータセットを作成する場合，入出力をコントロールできないため，入力データから出力テキストを１つに推定できない問題も存在する．実世界で得られた時系列数値データと，そのデータに関して実際に逐次配信されたテキストを用いて構築された既存のデータセットにおける上記の問題について，分析，解決策を探り，目的のテキストを生成できるようにする．

## 1.4 本論文の構成

以下に本論文の構成を述べる．**第 2 章**では，本論文で用いるニューラルネットワークの基礎技術について，**第 3 章**では，言葉の意味を捉える技術について説明する．**第 4 章**から**第６章**では，言語と非言語の対応関係獲得課題について取り上げる．**第 4 章**では，言語から非言語の生成課題として，自然言語理解におけるロボットの動作生成について，**第５章**では，非言語から言語の生成課題として，一般ドメインの実況生成課題を取り上げる．**第６章**では，非言語から言語の生成課題で発生する問題のうち，実データを用いた data-to-text における問題点を上げ，それの解決策について述べる．最後に，**第７章**で，言語と非言語の対応関係獲得課題の結論を述べる．

# 第2章　ニューラルネットワーク

## 2.1　順伝播型ネットワーク

ニューラルネットワークは人間等の脳神経系を模倣したモデルである．ネットワークを構成する各ノードは図 2.1 のように複数の入力を受け取り，1 つの出力を計算する．

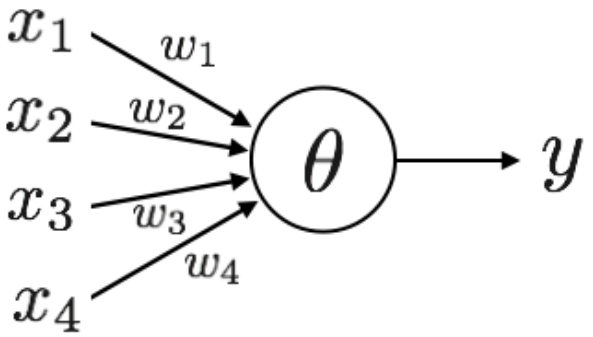



図 2.1: 各ノードの入出力

ここで，$w_i$ は $i$ 番目の入力 $x_i$ に対する結合の重み，$\theta$ はしきい値であり，出力 $y$ は

$$y = f\left(\sum_{i=1}^{4} w_i x_i - \theta\right) \tag{2.1}$$

と表される．ここで常に 1 の値を取る $x_0$ を導入し，$\boldsymbol{x} = (x_0, \cdots, x_4)$，$\boldsymbol{w} = (w_0, \cdots, w_4)$ とすると，

$$y = f\left(\sum_{i=0}^{4} w_i x_i\right) = f(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) \tag{2.2}$$

となる．ここで，$w_0 = -\theta$ であり，関数 $f$ は活性化関数である．

順伝播型ネットワークとは一般に，入力層，複数の隠れ層（中間層），出力層からなる，入力層から出力層への単一方向へ信号が伝搬されながら，式 2.2 に従って変換されるネットワークである．ノードは隣り合う層間でのみ結合し，それぞれの結合には結合の重みが割り当てられている．具体例を図 2.2 に示す．各層での出力を式で表すと，

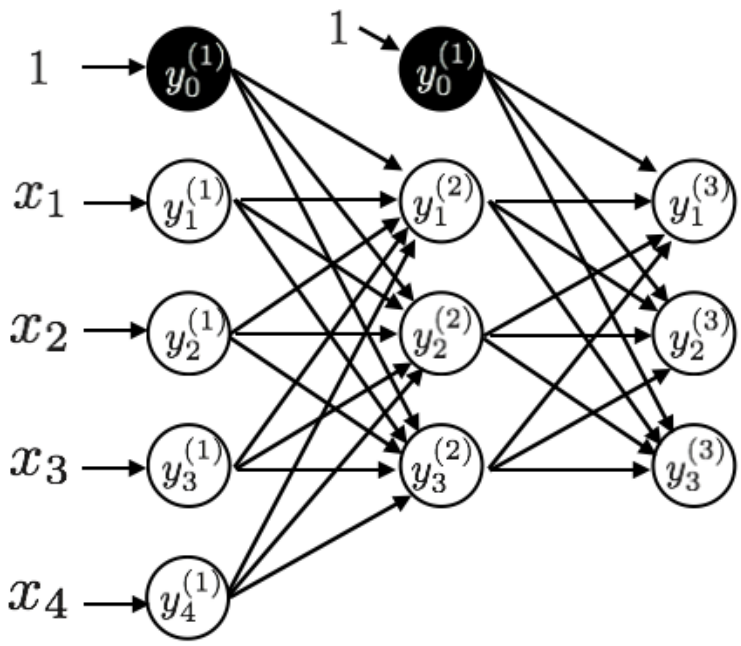



図 2.2: 3 層の順伝播型ネットワーク

$$y_j^{(2)} = f^{(1)}\left(\sum_{i=0}^{4} w_{ij} y_i^{(1)}\right) = f^{(2)}(\boldsymbol{w_j}^{\mathrm{T}} \boldsymbol{y^{(1)}}) \tag{2.3}$$

$$y_k^{(3)} = f^{(3)}\left(\sum_{j=0}^{3} w_{jk} y_j^{(2)}\right) = f^{(3)}(\boldsymbol{w_k}^{\mathrm{T}} \boldsymbol{y^{(2)}}) \tag{2.4}$$

となる．第 $u-1$ 層のノード数を $i = 1, \cdots, I$，第 $u$ 層を $j = 1, \cdots, J$ としたとき，第 $u$ 層での出力 $y$ は次のように一般化される．

$$y_j^{(u)} = f^{(u)}\left(\sum_{i=0}^{I} w_{ij} y_i^{(u-1)}\right) = f^{(u)}(\boldsymbol{w_j}^{\mathrm{T}} \boldsymbol{y^{(u-1)}}) \tag{2.5}$$

ネットワークの結合の重みをまとめて $\boldsymbol{W}$ とすると，順伝播型ネットワークの出力は，$\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{W})$ のように入力ベクトルと結合の重みによって決定される．

### 2.1.1 活性化関数

活性化関数 $f$ には一般に単調増加する非線形関数が用いられる．

$$f(x) = (1 + \exp(-x))^{-1} \tag{2.6}$$

$$f(x) = \tanh(x) \tag{2.7}$$

のようなロジスティック関数や双曲線正接関数といったシグモイド関数が主に使われるが，計算コストが大きく，また深いニューラルネットの学習では誤差消失がおこるため，近年では ReLU (Nair and Hinton, 2010)

$$f(x) = \max(x, 0) \tag{2.8}$$

や Softplus 関数

$$f(x) = \log(1 + \mathrm{e}^{x}) \tag{2.9}$$

が用いられることがある．

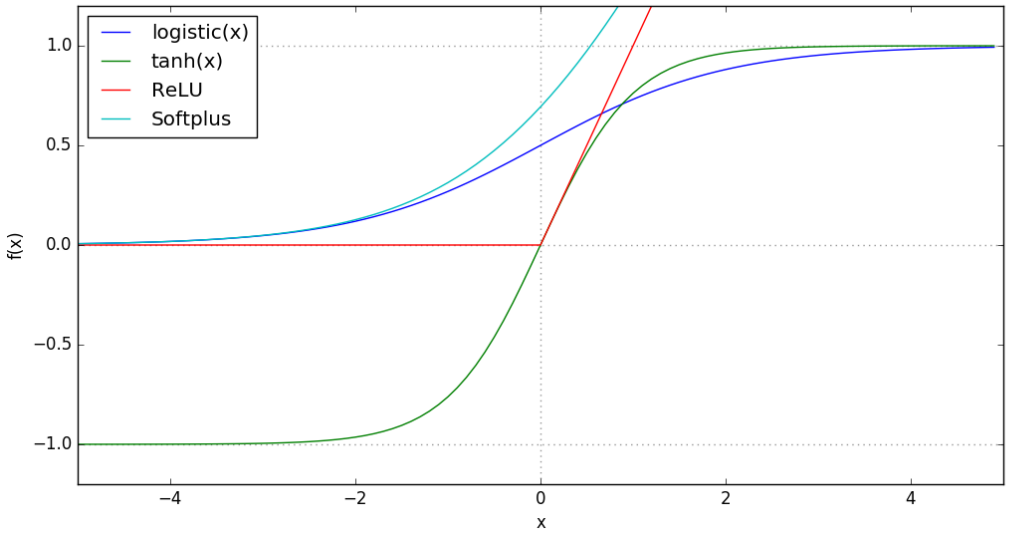



図 2.3: ニューラルネットワークで用いられる代表的な活性化関数

### 2.1.2 出力層の設計

出力層での活性化関数は問題の種類により適切なものを使用する．回帰問題では目的関数が任意の実数の場合は恒等写像を，値域が $[-1:1]$ の場合は双曲線正接関数を用いる．二値分類にはロジスティック関数を，多クラス分類には出力の合計が 1 となるソフトマックス関数を用いることが多い．

### 2.1.3 誤差逆伝播法

順伝播型ネットワークの学習は与えられた訓練データと出力された結果との誤差を最小化することである．つまり，誤差関数 (損失関数)$\boldsymbol{E} = E(O_0^{(m)}, \cdots, O_K^{(m)})$ の勾配を計算し，重み $\boldsymbol{w}$ について最小化することで目的関数を導出する．$m$ 層のネットワークを考える．$k$ 層の第 $i$ ノードへの入力の総和を $I_j^{(k)}$，出力を $O_j^{(u)}$，$k-1$ 層の第 $i$ ノードから $k$ 層の第 $j$ ノードへの結合の重みを $w_{ij}^{(k)}$，活性化関数を $f^{(k)}(\cdot)$ とすると，これらの関係は以下となる．

$$
\begin{aligned}
I_j^{(k)} &= \sum_{i=0}^{I} w_{ij}^{(k)} O_i^{(k-1)} && (2.10)\\
O_j^{(k)} &= f^{(k)}(I_j^{(k)}) && (2.11)
\end{aligned}
$$

誤差関数 $\boldsymbol{E}$ を $w_{ij}^{(k)}$ について微分すると，

$$
\begin{aligned}
\frac{\partial \boldsymbol{E}}{\partial w_{ij}^{(k)}} &= \frac{\partial \boldsymbol{E}}{\partial O_j^{(k)}} \frac{\partial O_j^{(k)}}{\partial I_j^{(k)}} \frac{\partial I_j^{(k)}}{\partial w_{ij}^{(k)}}\\
&= \frac{\partial \boldsymbol{E}}{\partial O_j^{(k)}} \frac{\partial f^{(k)}(I_j^{(k)})}{\partial I_j^{(k)}} \frac{\partial (\sum_{i=0}^{I} w_{ij}^{(k)} O_i^{(k-1)})}{\partial w_{ij}^{(k)}}\\
&= \frac{\partial \boldsymbol{E}}{\partial O_j^{(k)}} \left\{ f^{(k)}(I_j^{(k)}) \right\}' O_i^{(k-1)} && (2.12)
\end{aligned}
$$

ここで

$$
d_j^{(k)} = \frac{\partial \boldsymbol{E}}{\partial O_j^{(k)}} \left\{ f^{(k)}(I_j^{(k)}) \right\}' \qquad (2.13)
$$

とする．$k = m$ 層以外の場合，さらに，

$$
\begin{aligned}
\frac{\partial \boldsymbol{E}}{\partial w_{ij}^{(k)}} &= \frac{\partial \boldsymbol{E}}{\partial O_j^{(k)}} \left\{ f^{(k)}(I_j^{(k)}) \right\}' O_i^{(k-1)}\\
&= \sum_l \frac{\partial \boldsymbol{E}}{\partial O_l^{(k+1)}} \frac{\partial O_l^{(k+1)}}{\partial I_l^{(k+1)}} \frac{\partial I_l^{(k+1)}}{\partial O_j^{(k)}} \left\{ f^{(k)}(I_j^{(k)}) \right\}' O_i^{(k-1)}\\
&= \left( \sum_l \frac{\partial \boldsymbol{E}}{\partial O_l^{(k+1)}} \left\{ f^{(k+1)}(I_l^{(k+1)}) \right\}' w_{jl}^{(k+1)} \right) \left\{ f^{(k)}(I_j^{(k)}) \right\}' O_i^{(k-1)}\\
&= \left( \sum_l d_l^{(k+1)} w_{jl}^{(k+1)} \right) \left\{ f^{(k)}(I_j^{(k)}) \right\}' O_i^{(k-1)} && (2.14)
\end{aligned}
$$

と変形出来る．以上より，

$$d_j^{(m)} = \frac{\partial \boldsymbol{E}}{\partial O_j^{(m)}} \left\{ f^{(m)}(I_j^{(m)}) \right\}' \tag{2.15}$$

$$d_j^{(k)} = \left( \sum_l d_l^{(k+1)} w_{jl}^{(k+1)} \right) \left\{ f^{(k)}(I_j^{(k)}) \right\}' \tag{2.16}$$

$$\frac{\partial \boldsymbol{E}}{\partial w_{ij}^{(k)}} = d_j^{(k)} O_i^{(k-1)} \tag{2.17}$$

によって誤差が出力層から入力層に向かって伝播する．

### 2.1.4 最適化手法

伝播した誤差は勾配降下法

$$w^{(t+1)} = w^{(t)} - \varepsilon \nabla \boldsymbol{E} \tag{2.18}$$

を用いて更新する．ここで $\varepsilon$ は学習率と呼ばれ，現在の重み $w^{(t)}$ を負の勾配方向 $-\varepsilon \nabla \boldsymbol{E}$ へ動かすことで重みを更新してより小さな $\boldsymbol{E}$ を求める．$\varepsilon$ が小さすぎると反復回数が増え学習にかかる時間が増大する．一方，$\varepsilon$ が大きすぎると $\boldsymbol{E}$ が増大する可能性がある．適切な学習率を選ぶことがニューラルネットワークの学習では重要である．

#### 確率的勾配降下法

確率的勾配降下法とは，データの一部を使って重み更新を行う勾配降下法である．すべての訓練データに対する誤差関数 $\boldsymbol{E}(w) = \sum_{n=1}^{N} E_n(w)$ を最小化するバッチ学習にくらべ，確率的勾配降下法は訓練データに冗長性がある場合などでより速く処理出来る．また，バッチ学習の場合は誤差関数 $\boldsymbol{E}(w)$ は常に同じである一方，確率的勾配降下法を用いると更新毎に異なる $E_n(w)$ のため，望まない局所最適解に収束することも少なく，より適切に学習が進む．

#### Adam

Adam (Kingma and Ba, 2015) は学習率を更新する手法の 1 つであり，勾配の平均と分散をオンライン推定した情報を利用する．以下にアルゴリズムを示す．

**Algorithm 1** Adam による学習率最適化

**Require:** $\alpha$: stepsize
**Require:** $\beta_1$,$\beta_2 \in$ [0,1): Exponential decay rates for the moment estimates
**Require:** $f(\theta)$: Stochastic objective function with parameters θ
**Require:** $\theta_0$: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  **while** $\theta_t$ not converged **do**
    $t \leftarrow t + 1$ (Initialize timestep)
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\hat{m}_t \leftarrow m_t/(1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\hat{v}_t \leftarrow v_t/(1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
  **end while**
  **return** $\theta_t$ (Resulting parameters)

## 2.2 再帰型ニューラルネットワーク

Recurrent Neural Network（再帰型ニューラルネットワーク，RNN）は過去の情報を一時的に記憶できるようなループ構造を持ち，時系列情報および可変長の入出力に対応できる特殊なニューラルネットワークであり，系列データを扱う自然言語処理や音声認識の分野で広く用いられる．RNN が持つ基本的な構造は入力層，中間層，出力層の 3 層からなる順伝搬型 NN と同様のものであるが，中間層に有向閉路，すなわちループ構造を持ち，その結合についての重みパラメータを持つ点が大きな違いである．

RNN は入力系列データ $\boldsymbol{x} = x_1, x_2, ..., x_T$ について，各時刻 $t$ につきひとつの入力 $x_t$ を受け取り，同時に系列出力 $y_t$ を逐次算出する．そのとき，中間層は入力層からの出力に加え，ループ構造によって一つ前の時刻 $t-1$ における中間層の出力を受け取り，計算を行う．同様に，時刻 $t-1$ の計算には時刻 $t-2$ の情報が再帰的に考慮されているため，理論上はそれまでに受け取った全ての時刻における入力 $\boldsymbol{x_t} = x_1, x_2, ..., x_{t-1}$ を反映して $y_t$ が計算されることになる．中間層における処理は，$W^{hx}$ を各時刻における入力 $\boldsymbol{x}_t$（$t = 1, ..., T$）に対する結合の重み，$W^{hh}$ を各時刻において一時刻前の中間層の出力 $\boldsymbol{h}_{t-1}$（$t = 1, ..., T$）に対する結合の重み，中間層の活性化関数を $f(\cdot)$ とすると，

$$\boldsymbol{h}_t = f(W_h h_{t-1} + W_x x_t) \qquad (t = 1, ..., T) \tag{2.19}$$

$$\boldsymbol{h}_0 = \boldsymbol{0} \ \textit{or initialized randomly} \tag{2.20}$$

のように表せる．$\boldsymbol{h}_0$ は $\mathbf{0}$，あるいは乱数で初期化されることが多いが，タスクに応じて特定の数値をおくこともある．図 2.4 に処理の流れを示す．

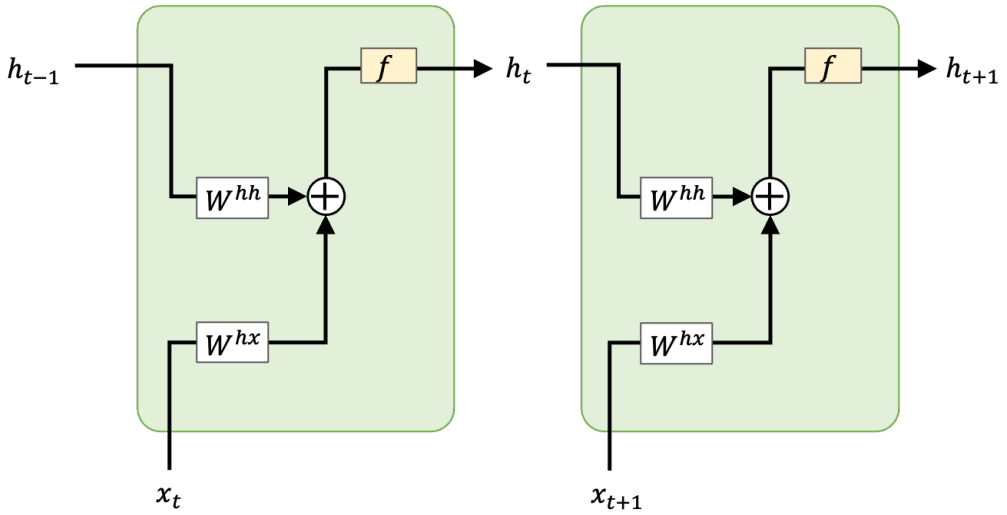



図 2.4: RNN の構造と処理の流れ

自然言語文生成に RNN を用いるときは，文を単語の系列データ（あるいは文字の系列データ）として表し，過去の文脈から $t$ 番目の単語を予測する．直前時刻の隠れ状態（時刻 1 から $t-1$ の単語情報），直前時刻の予測結果（時刻 $t-1$ の単語）を入力とし，逐次的に次の単語の予測を繰り返して文章を生成する．文生成を行う RNN は RNN 言語モデルと呼ばれ，キャプション付けや機械翻訳のタスクで用いられている．

このように順伝搬型ネットワークとは異なる特性と構造を持つ RNN であるが，学習そのものは順伝搬型ネットワークとほとんど同様のアルゴリズムで行うことができる．中間層の出力は時刻毎に変動し次時刻に影響を与えるものの，重みを用いる乗算総和処理については入力系列データそれぞれの要素 $x_t$ について同一の重みパラメータを用いるため，入出力長は可変長でありながら学習すべきパラメータは固定長となっている．また，ループ構造を系列の時刻方向に展開すると，通常の順伝搬型ネットワークの計算として一般化することができ，誤差逆伝搬法を含む学習アルゴリズムの適用が可能となる．このように RNN を学習する手法は Backpropagation Through Time（BPTT）法と呼ばれる．

しかし，標準的な RNN では，短期的な時系列情報は扱うことができても，長い時系列情報をうまく扱えないという問題が指摘されている (Bengio et al., 1994)．RNN の伝搬計算は時間方向に展開されるため，長すぎる系列情報を扱う RNN は深すぎるニューラルネットと同様に勾配消失問題などの技術的問題が発生してしまうことが大きな要因のひとつとされる．

そこで，Long Short-Term Memory（LSTM） や GRU に代表される，より長い系列の

予測や学習を行えるような複雑な構造を持つ RNN が考案され，広く使用されている．

### 2.2.1 LSTM

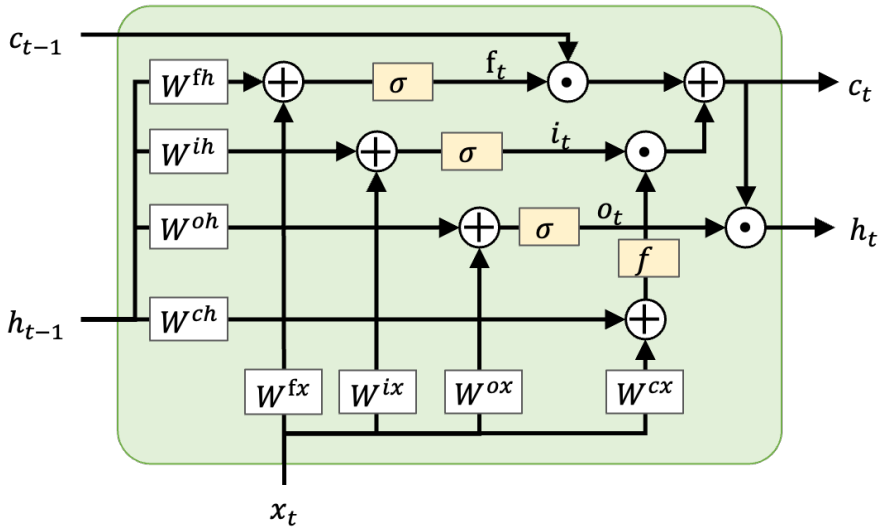



図 2.5: LSTM の構造と処理の流れ

LSTM の概要図を図 2.5 に示す．LSTM (Hochreiter and Schmidhuber, 1997) は RNN の中間層 1 層に相当する構造である．単純な RNN の中間層と異なり，LSTM は中間層内に過去の計算情報を隠れ状態として保持するメモリセル，および入出力の情報量やメモリセルの更新量を時刻毎に制御するゲートを持つ．直前時刻の計算結果として得られる $h_{t-1}$ だけでなく，それまでの計算情報をゲートによって重み付けして選択的に更新されるメモリセルの状態 $c_{t-1}$ を用いることで，初期の時刻の情報を長期間保てる機能を実現し，長期依存の学習を可能にしている．最初期の LSTM (Hochreiter and Schmidhuber, 1997) には忘却ゲートはなく，ゲートは入出力の 2 つしかなかったが，一般に忘却ゲートの導入によって性能向上が見込めるとされ，現在では入力ゲート，忘却ゲート，出力ゲートの 3 種類のゲートを用いることが多い (Gers et al., 1999)．以降，Gers et al. (1999) のモデルを説明する．

入力 $x_t$ および前時刻の出力 $h_{t-1}$ をゲートにそれぞれ受け取って，$[0, 1]$ の範囲の値をとり，そのゲートを通過する情報を反映させる量の割合を制御する．メモリセルの値および出力 $h_t$ は入力 $x_t$ と前時刻の出力 $h_{t-1}$ に加えて前時刻のメモリセルの値 $c_{t-1}$ から計算されるが，ゲートによってそれらの情報量が制限されるようになっている．入力をメモリセルおよび各ゲートに変換する重みパラメータを $W^{cx}, W^{ix}, W^{ox}, W^{fx}$，直前の隠れ状態をメモリセルおよび各ゲートに変換する重みパラメータを $W^{ch}, W^{ih}, W^{oh}, W^{fh}$ とし，時刻 $t$ における入力ゲート $i_t$，忘却ゲート $\mathrm{f_t}$，出力ゲート $o_t$，メモリセル $c_t$，出力 $\boldsymbol{h}_t$ は，以下のよう

に算出される．

$$
\begin{aligned}
i_t &= \sigma(W^{ih}h_{t-1} + W^{ix}x_t) && (t = 1, ..., T) && (2.21)\\
\mathrm{f}_t &= \sigma(W^{fh}h_{t-1}) + W^{fx}x_t)) && (t = 1, ..., T) && (2.22)\\
o_t &= \sigma(W^{oh}h_{t-1} + W^{ox}x_t) && (t = 1, ..., T) && (2.23)\\
c_t &= \mathrm{f}_t \odot c_{t-1} + i_t \odot f(W^{ch}h_{t-1} + W^{cx}x_t) && (t = 1, ..., T) && (2.24)\\
\boldsymbol{h}_t &= o_t \odot c_t && (t = 1, ..., T) && (2.25)\\
\boldsymbol{h}_0 &= \mathbf{0} \ \ or\ initialized\ randomly && && (2.26)
\end{aligned}
$$

なお，忘却ゲートの値 $\mathrm{f}_t$ と活性化関数 $f(\cdot)$ は便宜同じ文字で表記したが，両者は無関係である．

### 2.2.2 GRU

GRU の概要図を図 2.6 に示す．LSTM と比較すると，メモリセルが存在せず，構造がシ

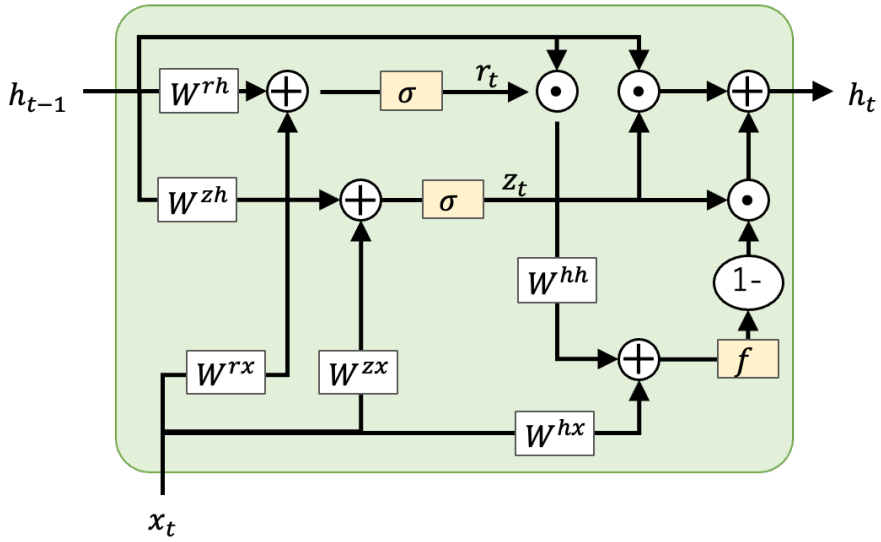



図 2.6: GRU の構造と処理の流れ

ンプルである．LSTM より少ないリセットゲートと更新ゲートの 2 つのゲートだけを使って，忘却と状態の更新を操作し，長期の情報は隠れ状態ベクトルに集約する．入力 $x$ を $r, z, h$ に変更する重みパラメータを $W^{rx}, W^{zx}$，$W^{hx}$ 直前の隠れ状態ベクトル $h$ を $r, z, h$ に変更する重みパラメータを $W^{rh}, W^{zh}, W^{hh}$ とすると，時刻 $t$ におけるリセットゲート $r_t$，更新

ゲート $z_t$，現時刻の隠れ状態であり，GRU の返り値 $h_t$ は，以下のように算出される．

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot f(W^{hh}(r_t \odot h_{t-1} + W^{hx}x_t) \quad (2.27)$$

$$r_t = \sigma(W^{rh}h_{t-1} + W^{rx}x_t) \quad (t = 1, ..., T) \quad (2.28)$$

$$z_t = \sigma(W^{zh}h_{t-1} + W^{zx}x_t) \quad (t = 1, ..., T) \quad (2.29)$$

LSTM と GRU には大きな精度の差はないが，GRU は LSTM よりもゲートが少なく，セルも不要のため，状態変数の数が同じであれば，GRU の方が LSTM より少ない計算量で済む．

## 2.3 Sequence-to-Sequence モデル

Sequence-to-Sequence モデル（系列変換モデル，seq2seq）は，系列（Sequence）を入力とし，別の系列（Sequence）を出力するモデルで，Sutskever et al. (2014) によって提案された．LSTM をはじめとした RNN では，エンコードした固定長のベクトルを出力することしかできず，長さがわからない入力を処理したり出力を生成することが難しい．そのため，例えば音声認識や翻訳に RNN をそのまま用いることは難しかった．Sutskever et al. (2014) の Sequence-to-Sequencen モデルでは，入力系列を LSTM でエンコードして固定長のベクトルに変形した後，そのベクトルを LSTM でデコードして出力系列を生成するため，RNN を系列変換にそのまま用いる場合の問題を解決できる．Sequence-to-Sequence モデルの発表と同時期に Cho et al. (2014) は同様のモデルを提案し，Encoder-Decoder モデルと呼んだ．近年では，入出力を系列に限らず，エンコーダー，デコーダーもそれぞれの入出力に合わせたネットワークを用いたモデルを，広く Encoder-Decoder モデルと呼ぶことが多い．また，Sequence-to-Sequence モデルのように，1 つのモデルでは表現しきれない処理をニューラルネットを用いてモデル化し，入出力情報だけを使って一気に学習する方法を，end-to-end モデルや end-to-end 学習と呼ぶこともある．本論文では，系列情報を扱うモデルでは Sequence-to-Sequence モデル，それ以外では Encoder-Decoder モデルと呼ぶこととする．

### 2.3.1 Attention 機構

Sequence-to-Sequence モデルでは，系列情報を固定帳のベクトルに変換するが，長い系列であるほど最初に入力された情報がデコーダーまで伝搬しにくい．そこで，注意機構 (Luong et al., 2015) を用いることで，より直接的に入力情報を出力に利用する．

長さ $I$ の入力系列 $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_I\}$ に対して，エンコードされたベクトルを $\boldsymbol{H}^{(e)} = \{\boldsymbol{h}_1^{(e)}, \ldots, \boldsymbol{h}_I^{(e)}\}$ とする．通常の Sequence-to-Sequence モデルでは，$\boldsymbol{h}_I^{(e)}$ をデコーダーの初

期隠れ状態 $\boldsymbol{h}_0$ として利用する．時刻 $t$ のデコーダーの隠れ状態 $\boldsymbol{h}_t$ を求める時，通常の Sequence-to-Sequence モデルの場合は，前時刻のデコーダーの隠れ状態 $\boldsymbol{h}_{t-1}$ と出力 $\boldsymbol{o}_{t-1}$ を用いて

$$\boldsymbol{h}_t = f(W_t[\boldsymbol{o}_{t-1}; \boldsymbol{h}_{t-1}]) \tag{2.30}$$

となる．この時，$f$ は活性化関数である．

注意機構を用いる場合はさらに，$\boldsymbol{h}_t$ とエンコーダーの隠れ状態 $\boldsymbol{H}^{(e)} = \{\boldsymbol{h}_1^{(e)}, \ldots, \boldsymbol{h}_I^{(e)}\}$ の関連度 $\boldsymbol{\alpha}_t = \{\alpha_{t,1}, \ldots, \alpha_{t,I}\}$ によって得られた，エンコーダーの隠れ状態の重みつき和である文脈ベクトル $\boldsymbol{c}_t$ を用いて，

$$\tilde{\boldsymbol{h}}_t = f(W_c[\boldsymbol{c}_t; \boldsymbol{h}_t]) \tag{2.31}$$

をデコーダの時刻 $t$ における最終的な隠れ状態とする．時刻 $t$ における，エンコーダーの隠れ状態 $\boldsymbol{h}_i$ の関連度を示すスカラー値の重み $\alpha_{t,i}$ と文脈ベクトル $\boldsymbol{c}_t$ は，次の式によって求められる:

$$\begin{aligned} \alpha_{t,i} &= \mathrm{align}(\boldsymbol{h}_t, \boldsymbol{h}_i^{(e)}) \\ &= \frac{\exp(\mathrm{score}(\boldsymbol{h}_t, \boldsymbol{h}_i^{(e)}))}{\sum_{i'=1}^{I} \exp(\mathrm{score}(\boldsymbol{h}_t, \boldsymbol{h}_{i'}^{(e)})} \end{aligned} \tag{2.32}$$

$$\boldsymbol{c}_t = \sum_{i=1}^{I} \alpha_{t,i} \boldsymbol{h}_i^{(e)}. \tag{2.33}$$

ここで，score(·) は以下の 3 種類のいずれかが用いられるが，その効果は大きくは変わらない：

$$\mathrm{score}(h_t, h_i^{(e)}) = \begin{cases} \boldsymbol{h}_t^\top \boldsymbol{h}_i^{(e)} & dot \\ \boldsymbol{h}_t^\top W_a \boldsymbol{h}_i^{(e)} & general \\ W_a[\boldsymbol{h}_t; \boldsymbol{h}_i^{(e)}] & concat \end{cases} \tag{2.34}$$

### 2.3.2 Copy 機構

Copy 機構は Gu et al. (2016) によって提案され，例えば文章要約タスクにおいて，辞書に含まれていない，入力文に出現した単語を要約文で生成したい場合など，入力側に含まれる要素をそのまま出力として利用したい場合に用いられる．具体的には，Attention 機構で計算される確率や，各入力要素がコピーされる確率を計算し，その確率に従って，入力要素をそのまま出力するかを決定する方法である．

## 2.4 Transformer

Transformer は Encoder-Decoder モデルをベースとするモデルであり，当初ニューラル機械翻訳のモデルとして提案された．Transformer が提案される以前は，機械翻訳などの課題には，LSTM や GRU に代表される RNN が多く使われていた．しかし，RNN は入力される系列を処理し，その処理した結果を用いて次の処理を行う，逐次的な処理を行っているため，並列化は不可能である．また，入出力系列の長さに関係なく，入力と出力の間の依存関係を得ることができる Attention 機構があるが，Attention 機構は RNN と組み合わせて使用されていることがほとんどであった．そこで，Vaswani et al. (2017) は，再帰性を除き，代わりに入力と出力の間のグローバルな依存関係を得るために，Attention 機構のみを使った Transformer を提案した．

### 2.4.1 Transformer の基礎構造

Transformer は Encoder と Decoder の両方に self-attention と Position-wise Feed-Forward Network が使われたモデルである．エンコーダ，デコーダとも入力系列を埋め込みベクトルに変換し，それに位置埋め込みベクトルを加えたものを入力とする．

**Encoder:** $N = 6$ の独立した層を持っており，各層には multi-head self-attention 構造と Position-wise Feed-Forward Network の 2 つの副層を持つ．2 つの副層それぞれを適用後には，残差結合と層正規化を行う．

**Decoder:** エンコーダと同じく，$N = 6$ の独立した層を持っており，エンコーダーの 2 つの副層に加えて，multi-head self-attention 構造の副層を 1 層追加した．各副層それぞれを適用後には，残差結合と層正規化を行う．

### 2.4.2 Transformer の要素

本項では，Transformer を構成する次の要素について説明する：1. Positional Encoding, 2. Multi-head self-attention, 3. Position-wise Feed Forward Network,

1. Positional Encoding
   入力の埋め込みベクトル（系列の要素数 $\times$ $d_{model}$）に位置情報（Positional Embedding）を付与する．要素の位置および入力の埋め込みベクトルの次元が一意に決まる位置情

報を入力の埋め込みベクトルに加算する．位置情報は下記の式で計算する．

$$PE(pos, 2i) = sin(pos/10000^{2i/d_{model}}) \tag{2.35}$$
$$PE(pos, 2i+1) = cos(pos/10000^{2i/d_{model}}) \tag{2.36}$$

$pos$ は要素が系列の何番目か，$2i$ は入力の埋め込みのベクトルの何次元目かを表す．例えば，300 次元の埋め込みにおいて，3 番目の要素の 13 次元目の位置情報は次のようになる：

$$PE(3, 13) = cos(3/10000^{12/300}). \tag{2.37}$$

2. Multi-head self-attention
Transformer で使われている Attention 機構は Query-Key-Value 型の Attention 機構である．Query と Key で Attention スコアを計算し，その Attention スコアを使って Value を加重和することで Attention を適用した要素の潜在表現を獲得する．Q と K は次元 $d_K$ を持ち，V は次元 $d_V$ を持つ．Transformer では縮小付きの Attention を採用しており，Attention の計算はクエリと全キーの内積によりその関連度を計算し，√ dK 割った後ソフトマックス関数を適用させる．Attention を使った層出力の計算は，以下の式で表される．

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.38}$$

   (a) self-attention
   Self-Attention とは，ある 1 文内の単語だけを使って計算された，単語間の関連度スコアのようなものである．このスコアを使って文のベクトル表現を獲得する．つまり，入力 (query) とメモリ (key, value) が同じものである．ある文に出現する単語間の関連度 (スコア，Attention) を計算するのにその文に出現する単語のみを使うことから，Self-Attention と言われている．

   (b) Multi-head attention
   Attention 機構を複数用いるため，次の式で表される Multi-head attention を用いる：

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2.39}$$
$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V). \tag{2.40}$$

   MultiHead の引数である $Q, K, V$ は系列の埋め込みベクトルを意味し，重みを掛ける前のものである．マルチヘッド注意機構では，自己注意機構における $Q, K, V$ をそれぞれ $h$ 個に分割して（系列の要素数 $\times$ $d_{model}/h$），それぞれ Scaled-Dot Product Attention を適用する．系列の埋め込みベクトルに $W_i^Q, W_i^K, W_i^V (i = 1, \dots h)$ を掛けることで $h$ 個に分割された行列を作成する．ここで，$W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in$

$\mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_k}$, $W^O \in \mathbb{R}^{d_{model} \times d_{model}}$ は重み行列である．また，$d_k = d_{model}/h$ である．Scaled-Dot Product Attention を計算してできた $head_i$ を $h$ 個分結合（concat）することで，self-attention 機構と同様の行列の形状（系列の要素数 $\times$ $d_{model}$）になる．

(c) Masked Attention
Decoder では Encoder での Self-Attention と異なり，ある時刻の入力から次の時刻の状態を予測し，さらにその予測を次の入力に回してその先を予測する．一方，学習のときにはすべての時刻で同時に次の時刻の要素を予測する学習を行うが，この時に Self-Attention で予測対象である未来の情報が得られると，先の答えが見えてしまうことになっている．そこで，Decoder では Self-Attention において Mask を行い，今までに得た情報のみで推論するように，先の attention weight を全て 0 にする．

3. Position-wise Feed Forward Network
2 層のネットワークからなっており，活性化関数は ReLU を使用する．全結合層に入力する行列を $x$ とし，$W_1$, $b_1$, $W_2$, $b_2$ は学習パラメータである．

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{2.41}$$

# 第3章　言葉の意味表現

現在，コンピュータで人の言葉を理解させる取り組みは様々行われており，身近な例ではかな漢字変換や検索などがあげられる．文章の意味理解には，一般的に構造的意味と語彙的意味を組み合わせて用いる．構造的意味とは文章の中で単語が組み合わされ表現される意味であり，構文解析を用いる．一方，語彙的意味とは単語が持っている元々の意味であり，WordNet などのシソーラス・オントロジーを用いる．しかし，大量の文章を処理する際にそれらを用いるのは計算コストが高いと考えられるため，また，Bag-of-Words 表現を用いて文章を理解する取り組みがあり，特に，単語をベクトルとして表すため BoW 表現を更に発展させ，単語の意味を分散表現で表した Word2vec が提案されたことで，コンピュータで自然言語を利用する手法が大きく発展した．近年は，文脈を考慮した単語の埋め込み表現も提案され，このような埋め込み表現を大規模なコーパスで学習した事前学習済みモデルや，事前学習済みモデルを使って，目的のタスクで高い性能を引き出すアプローチ（ファインチューニング）が提案されている．

## 3.1　分散意味表現

分散表現とは，Hinton (1984) によって提案された単語を固定長の高次元の実数ベクトルで表現し，類似の意味を持つ単語を近いベクトルに対応させる技術である．各概念に一つの計算要素を割り当てる局所表現 (one-hot representation) に比べ，より単語同士の関係とベクトル同士の関係が対応づけられる．ベクトル構成方法は大きく 2 つのアプローチがある．

1 つは BoW の考えを用い，単語の意味はその単語が出現する文脈によって決まるという分布仮説 (distributional representation) (Harris, 1954; Firth, 1957) を元にコーパス内の単語の出現頻度，単語周辺の文脈や共起などの統合情報をまとめた行列を作成，次元削減しベクトルを求める手法で，分布意味表現 (Distributional Semantic Representations) と呼ばれる．共起頻度の重み付けに相互情報量を用いた PPMI(Positive Pointwise Mutual Information) に基づく単語文脈行列 (Bullinaria and Levy, 2007) や，次元削減に特異値分解を用いた LSI(Latent Semantic Indexing)(Deerwester et al., 1990) が知られている．

一方，分布仮説を用いるが各単語の意味表現が $m$ 次元で表されると仮定しベクトルの要素をコーパスから学習する手法があり，Word embedding や分散意味表現 (Distributed Semantic Representations) と呼ばれる．分布意味表現に比べ，より低次元で密に表すこと

ができ，更にベクトルは有数の次元である．次節より分散意味表現を求める手法について説明する．

## 3.2 ニューラルネットワーク言語モデル

Bengio et al. (2003) が提案したニューラルネットワーク言語モデル (NNLM) は，Shannon (1948) によって提案された N-gram 言語モデルをニューラルネット化したモデルであり，語彙数を $V$ とする連続する単語列 $w_1, \ldots, w_T$ が存在し，単語列 $w_{t-1}, \ldots, w_{t-n+1}$ が与えられたとき，次の単語 $w_t$ の出現する条件付き確率を出力するニューラルネットワークである．単純な BoW 表現は語順を考慮しないが，NNLM は N-gram を用いるため語順情報を得ることができる．図 3.1 に概要を示す．

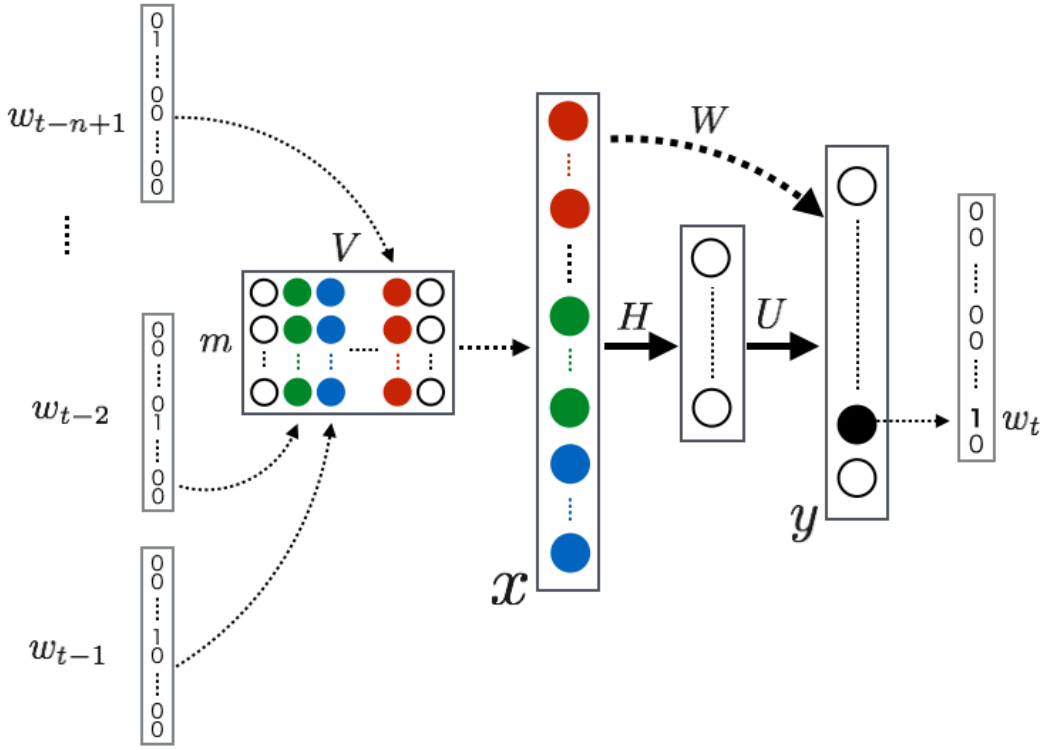



図 3.1: ニューラル言語モデル

各単語は 1-of-N coding により表現されており，look-up テーブル $C$ 上の $C(i)$ にそれぞれ m 次元のベクトルを持つ．$n-1$ 個の単語 $w_{t-1}, \ldots, w_{t-n+1}$ それぞれに対するベクトル $C(w_{t-1}), \ldots, C(w_{t-n+1})$ を連結し，$(n-1)m$ 次元のベクトル $x$ が入力となる．このとき，中間層に対する重み行列を H，中間層でのバイアスを d，中間層での活性化関数を tanh 関数，中間層から出力層に対する重み行列を U，入力層から出力層に対する重み行列を W，出

力層でのバイアスを b とすると正規化前の出力 $y$ は

$$y = b + Wx + U \tanh(d + Hx) \tag{3.1}$$

と表される．よって，単語 $w_t$ の出現する条件付き確率は softmax 関数を用いて，

$$P(w_t|w_{t-1}, \ldots, w_{t-n+1}) = \frac{\exp(y_t)}{\sum_{i=1}^{V} \exp(y_i)} \tag{3.2}$$

となる．

## 3.3 Word2vec

Mikolov et al. (2013a) によって公開された Word2vec[1]は，同じ文脈の中にある単語はお互いに近い意味を持つようにニューラルネットワークを用いて単語の分散意味表現を得るツールであり，その学習方法として CBOW と Skip-gram の 2 つの手法を提案している．これにより，単語の意味はベクル計算によって表現可能となり，例えば ‘king’-‘man’+‘woman’ (「王様から男を引いて女を足したもの」) を求める計算をすると，単語 ‘queen(「女王」)’ のベクトルに近い値が得られる，といった単語の演算が可能になる．

### 3.3.1 CBOW

Continuous Bag-of-Words model(CBOW) とは，ある単語 $w_t$ が現れている文脈で周辺単語を使って $w_t$ を予測するモデルである．前の文脈から単語を予測する NNLM と違い，CBOW は後ろの単語列を用いることができる．つまり $w_t$ を予測する場合，直前の $n$ 個の単語 $w_{t-n}, \ldots, w_{t-1}$ と直後の $n$ 個の単語 $w_{t+1}, \ldots, w_{t+n}$ を特徴量として用いることができる．

CBOW は単語 $w_t$ が $t$ 番目に出現する確率 $p(w_t|w_{t-n}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+n})$ を学習するので文脈は $2n$ 個の単語からなり，例えば $w_t$ の文脈表現 $\hat{\boldsymbol{w}}_t$ を文脈中の単語ベクトル平均とすると，

$$\hat{\boldsymbol{w}}_t = \frac{1}{2n}(\boldsymbol{w}_{t-n} + \cdots + \boldsymbol{w}_{t-1} + \boldsymbol{w}_{t+1} + \cdots + \boldsymbol{w}_{t+n}) \tag{3.3}$$

と表すことができ，図 3.2 で示される CBOW は，

$$p(w_t|w_{t-n}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+n}) = \frac{\exp({\hat{\boldsymbol{w}}_t}^{\mathrm{T}} \boldsymbol{w}_t)}{\sum_{i=1}^{V} \exp({\hat{\boldsymbol{w}}_i}^{\mathrm{T}} \boldsymbol{w}_i)} \tag{3.4}$$

[1] https://code.google.com/p/word2vec/

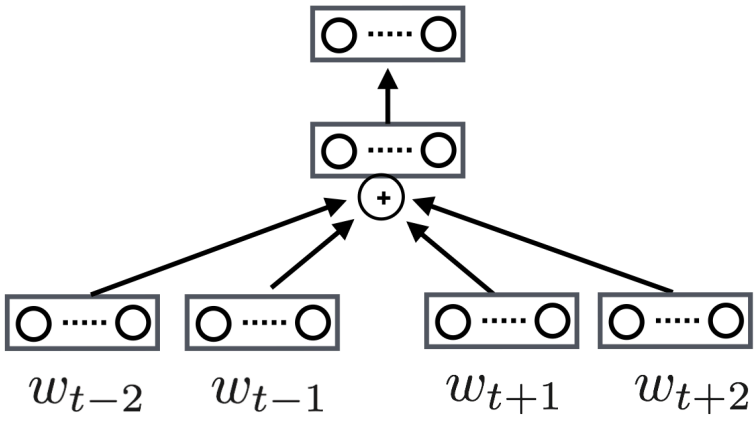



図 3.2: CBOW

のように表される．ここで $V$ は語彙数である．この場合，NNLM の中間層での活性化関数を $+$ 演算子として置き換えたものと考えることができる．

なお，$w_t$ の文脈表現 $\hat{\boldsymbol{w}}_t$ を式 3.3 とすると単語の語順を考慮することができないので，出現位置に考慮した重み付けや $r$ 次元の文脈ベクトルを $2n$ 個を連結し $2nr$ 次元のベクトルと表現することも可能であるが，パラメータの調節が増える．

### 3.3.2 Skip-gram

周辺単語から単語を予測する CBOW と違い，Continuous Skip-gram Model(Skip-gram) はある単語 $w_t$ から周辺単語を予測するモデルであり，図 3.3 のように示される．

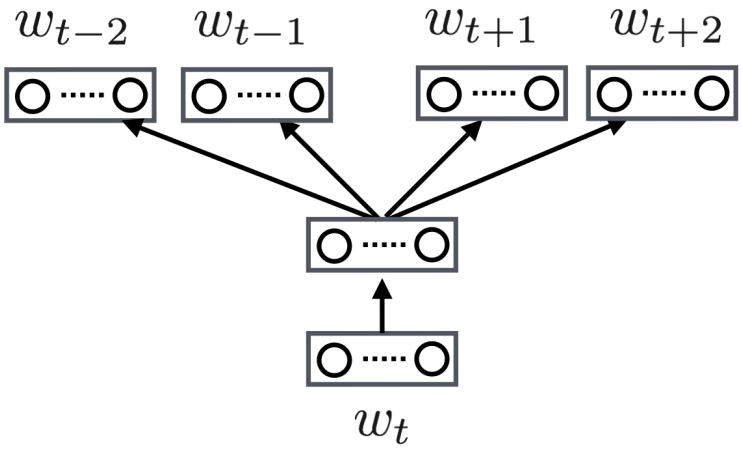



図 3.3: Skip-gram

ある単語 $w_t$ の文脈中に出現する別の単語 $w_j$ の出現確率 $p(w_j|w_t)$ は log-bilinear model を用い，

$$p(w_j|w_t) = \frac{\exp(\boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{w}_j)}{\sum_{i=1}^{V(w_t)}\exp(\boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{w}_i)} \tag{3.5}$$

と定義される．ここで $V(w_t)$ は単語 $w_t$ が含まれる文脈中に出現する語彙の集合である．

単語 $w_t$ から周辺単語列 $C_{w_t} = w_{t-n}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+n}$ を予測するモデルは，

$$p(C_{w_t}|w_t) = \prod_{-n\leq i\leq n, i\neq 0} p(w_{t+i}|w_t) \tag{3.6}$$

$$\log p(C_{w_t}|w_t) = \sum_{-n<i<n, i\neq 0} \log p(w_{t+i}|w_t) \tag{3.7}$$

となり，目的関数

$$J = \frac{1}{T}\sum_{t=1}^{T}\sum_{-n<i<n, i\neq 0} \log p(w_{t+i}|w_t) \tag{3.8}$$

$$p(w_{t+i}|w_t) = \frac{\exp(\boldsymbol{w}_{t+i}^{\mathrm{T}}\boldsymbol{w}_t)}{\sum_{j=1}^{V(w_t)}\exp(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{w}_t)} \tag{3.9}$$

を最大化するように学習をする．なお式 3.9 は $p(w_{t+i}|w_t)$ について式 3.5 のベクトルの内積を交換法則を用いて変形したものである．式 3.9 の分母はコーパスに含まれる全ての単語に関する内積を求めるため計算コストが高い．そこで Mikolov et al. (2013b) はロジスティック回帰を用いて近似し，更に負例 $S(w_{t+j})$ はランダムに選択する Negative Sampling を提案した．Negative Sampling により，式 3.9 は

$$p(w_{t+i}|w_t) = \frac{\exp(\boldsymbol{w}_{t+i}^{\mathrm{T}}\boldsymbol{w}_t)}{\sum_{j=1}^{V(w_t)}\exp(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{w}_t)} \approx \sigma(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{w}_t)\prod_{\boldsymbol{w}_i\in S(w_{t+j})}\sigma(-\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{w}_i) \tag{3.10}$$

のようになる．

## 3.4 事前学習済みモデル

単語埋め込みは，単語に関する一般的な知識を表現できると期待され，さまざまな言語処理タスクで利用された．しかし，word2vec などの単語埋め込みは文脈から切り離されており，文脈を考慮すべき課題では，RNN などでネットワークを構築し，教師あり学習を行うことで，課題に応じた訓練を行う必要があった．そこで，文脈における単語の意味を表現できる単語埋め込みを汎用的に学習する手段として，双方向 LSTM を用いて言語モデルを学習することで，文全体を考慮した単語の意味表現ベクトルを獲得できる ELMo (Peters et al.,

2018) などが提案された．ELMo などによって得られた単語ベクトルは, contextualized word embedding（文脈化単語埋め込み）と呼ばれる．単語埋め込み表現を学習するモデルと，それを利用する課題のモデルは当初，個別に構築されていた．このため，課題に利用される埋め込み表現は，事前学習された埋め込み表現のみで，その学習モデルの内部状態を利用することはなかった．これに対し，埋め込み表現を学習するモデルと，それを利用する課題を同一の構成とし，事前学習された埋め込み表現だけでなく，事前学習に用いたモデルのパラメータを課題にそのまま利用し，そのパラメータを課題の訓練データで微調節するファインチューニングを行う手法が提案され，その有用性が示されている．

### 3.4.1 BERT

ELMo では 3 層の LSTM が用いられているが，その後 12 層の Transformer のデコーダー部分を採用し, 大規模なコーパスを用いて事前学習した GPT (Radford et al., 2018) が提案された．GPT では Transformer のデコーダー部分を利用したため, 事前学習では前の文脈を参照し, 次の単語を予測するタスクにしか取り組んでいない．そこで, Transformer のエンコーダー部分を用いて，双方向から文を参照する事前学習タスクを提案したのが BERT (Devlin et al., 2019) である．BERT は任意の位置の単語をマスクし，そのマスクされた単語を予測するタスクを事前学習に用いることで，文全体を参照する言語モデルを学習する．さらに, BERT は入力文がコーパス中における隣接文どうしかを識別するタスクを事前学習に用いたことで，文の埋め込み表現を獲得することを目指した．図 3.4 に BERT のモデル構造を示す．

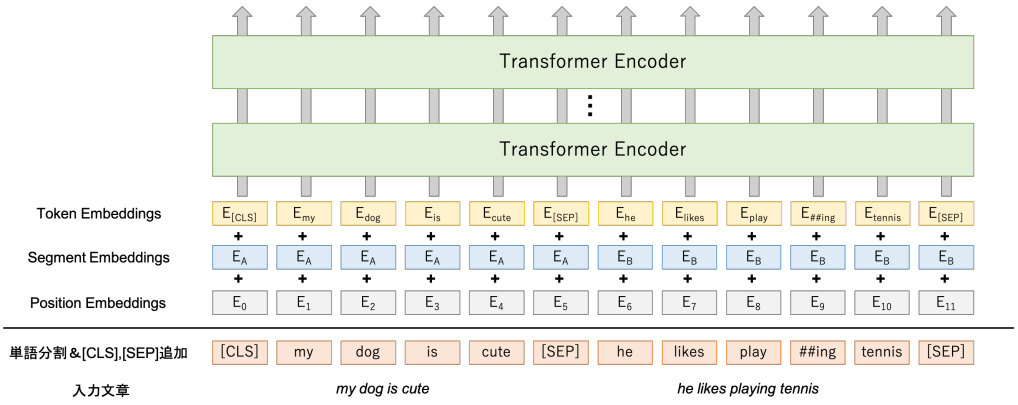



図 3.4: BERT の構造および入力形式

### BERT の構造

BERT は Transformer のエンコーダ部分を基とする構造を使用している．BERT には $\text{BERT}_{\text{base}}$ と $\text{BERT}_{\text{large}}$ が存在する．L は Transformer のエンコーダ層の層数，H は隠れ層における次元（単語埋め込みの次元），A は Multi-head self-attention 機構におけるヘッド（2.4 節の Multi-head self-attention 機構における $h$ に相当）の数とすると，$\text{BERT}_{\text{base}}$ は L=12，H=768，A=12，パラメータ総数=110M，$\text{BERT}_{\text{large}}$ は L=24，H=1024，A=16，パラメータ総数=340M である．FFN 層における重みは $W_1 \in \mathbb{R}^{H\times 4H}$, $W_2 \in \mathbb{R}^{4H\times H}$ である．

### 入出力の形式

BERT ではファインチューニングの際に様々な下流タスクに対応できるよう，トークンが連続している単文または文ペアを入力することができる．また，入力は文ではなく任意の長さのテキストでも入力が可能である．入力形式は，文の初めは必ず special classification token（[CLS]）である．また，文末には special token（[SEP]）を入力し，文ペアを入力する時も，この [SEP] で区切って入力する．これらの BERT への入力トークン列をシーケンスと呼ぶ．BERT に入力されたシーケンスは図 3.4 のように，Token Embeddings，Segment Embeddings，Position Embeddings の和をとることで，埋め込み表現に変換される．

### 事前学習タスク

BERT はマスクしたトークンを予測する Masked Language Mode と，文章の連続性を判定する Next Sentence Prediction の 2 つのタスクを使って，事前学習する．

**Masked Language Mode**　MLM は入力トークンの約 15%のトークンをマスクし（入力トークンを [MASK] トークンに置き換えて隠す），マスクしたトークンを予測するタスクである．マスクされているトークンは，直前の単語からのみではなく，文脈を考慮して文全体の入力から予測される．また 15%のトークン全てを [MASK] トークンに置き換えずに，15%選んだトークンのうち，80%のトークンを [MASK] トークンに置き換え，10%の単語はランダムなトークンに置き換え，残りの 10%のトークンは置き換えない，という手法をとる．選ばれた 15%のトークンは，最終層から出力された特徴量を用いてクロスエントロピーを誤差として学習する．

**Next Sentence Prediction (NSP)**　NSP は文のペアを入力し，その文ペアが連続している文であるかを予測する 2 値分類タスクである．質問応答タスク（QA）や自然言語推論タスク（NLI）では文ペアの関係性が重要であるため，NSP タスクを事前学習タスクとし

て利用する．ここでは入力として，“関係がなく文脈の繋がりがない文ペア” と “連続的に存在していて意味に繋がりがある文ペア” の 2 パターンを用意する．このタスクでは，文頭に入力した [CLS] に対する最終層の出力を用いて，2 値分類タスクを解く．

### 3.4.2 BART

BART (Lewis et al., 2020) は，Transformer のエンコーダ・デコーダの両方を備えた事前学習済みモデルである．Transfomer のデコーダのみで構成された GPT と比較すると，入力文に強く条件づけられた言語生成タスクと相性が良いとされている．

#### BART の構造

BERT，GPT および BART の比較のための概要を図 3.5 に示す．BERT では，トークンの一部を [MASK] に置き換え，文をエンコードする．Mask されたトークンは独立して予測されるため，BERT を生成に使うのは難しい．GPT では，トークンを自動的に再起的に予測するため，生成に使うことができる．しかし，ある時刻から次の時刻の状態を予測するデコーダのため，双方向の文脈を学習できていない．そこで，2 つの機能を併せ持つ，BART が提案された．図 3.5 の BART では，トークンの一部を [Mask] で置き換えられた文は，双方向の文脈を参照できる Transformer のエンコーダでエンコードされ，自己回帰のデコーダーにより，元の文章を生成している．このように BERT とは異なり，文章を生成することができるため，入力と出力の長さを変えることができ，さらに事前学習の方法に自由度がある．

#### BART の事前学習

BART では，さまざまな文章にノイズを付加し，再構成損失（デコーダの出力と元の文書との間のクロスエントロピー）を最適化する学習方法で事前学習を行う．ノイズ付加の方法は BERT と同様の手法も含まれるが，新たな付加手法も提案している．

**Token Masking**　BERT と同様にランダムに単語を [MASK] で置き換える．

**Token Deletion**　ランダムに単語を削除し，その単語を埋めた文章を生成することを学習する．[MASK] で置き換える場合と異なり，どの単語が抜けているかわからないので，場所も含めて予測する必要があり，Token Masking と比べて難易度が上がる．

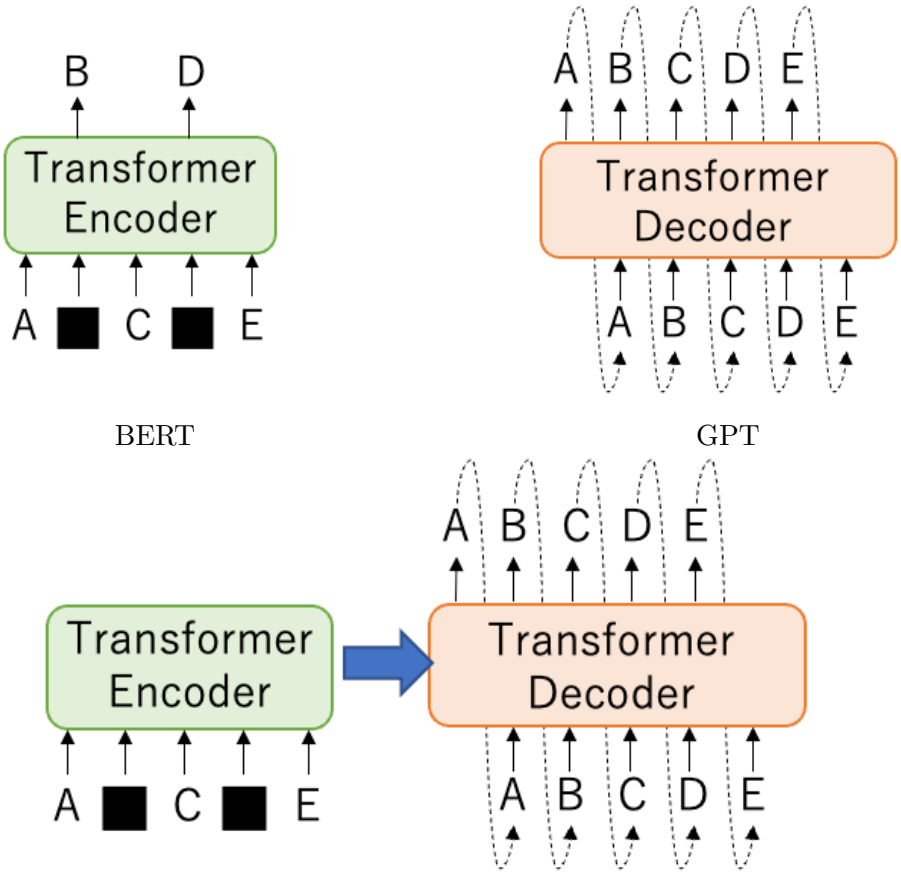



図 3.5: BART と BERT および GPT の比較

**Text Infilling**　複数の単語の並びを一つの [MASK] で置き換える．置き換える単語数は、$\lambda = 3$ のポアソン分布に従う乱数を発生させて決定するため，平均 3 個 (分散も 3) の連続した単語が [MASK] に置き換わる．

**Sentence Permutation**　複数の文章からなる文書の，文章の順番を入れ替える．

**Document Rotation**　文章から単語を一つ選び，その単語が一番初めになるように文章を回転させる．例えば，元の文章が「私　は　犬　を　飼って　いる」であれば，「犬　を　飼って　いる　私　は」のように回転させ，どの単語が初めの単語であるかを学習する．

# 第4章　自然言語理解によるロボットの動作生成

本章では，言語から非言語の生成課題の１つとして，自然言語理解によるロボット動作の生成課題を扱う．

## 4.1　研究背景と目的

現在，日本は超高齢化社会に突入している．少子高齢化社会の到来による人手不足を，ロボットを利用することで問題解決をはかろうとする場面が増えると考えられる．近年では，ロボットを安価に入手出来るようになっており，またロボットに触れる機会も増えたことにより，人間とロボットとのコミュニケーションが大きく発展する可能性がある．家庭内でロボットを用いる場合，ロボットが居住者と協調して暮らすことができる条件として，言葉や身振りを用いいることで居住者の経験をロボットに伝え，ロボットはそれを真似し，学習することが必要になると考える．このことを踏まえ，人の言葉による指示からロボットが動作を生成し動作することを目標にする．

まず，ロボットが人の動作を真似て行うことができるように，自らが行える基本動作を組み合わせてひとつの行為を生成する枠組みを構築し，複数の基本動作から複雑な行為を表現する対応関係を明確にする．

また，言葉と動作の対応関係を学習することによって，初めて行う動作であっても言葉の意味から推測し，動作を行なえるようにし，多様なロボット動作と多様な人の言葉の対応関係を学習する枠組みを提案し検証する．なお，本章ではロボットの動作として調理に関する動作を対象とし，家庭内でロボットへの指示に慣れていないユーザーが使用することを想定する．そのため与える言葉の指示は，明確な指示，例えば「10cm 右手を動かす」や「秒速 1m で動く」などの言葉ではなく，「大きく」など，その対象や状況によって量や数が変化する，曖昧な言葉の表現に注目する．このような言葉の曖昧さを許容し，ロボット動作と言葉の対応関係を得る枠組みを提案する．

## 4.2 関連研究

ニューラルネットワークやリカレントニューラルネットワーク (RNN) を用い，ロボットに動作を生成させる研究は広く行われている．

神経力学モデル MTRNN(Multiple Timescale Recurrent Neural Network)(Yamashita and Tani, 2008) は RNN を拡張したモデルで，再起結合を持つことで力学モデルとしての性質を持つ．現在の状態を入力とし，次時刻の動作を出力とするものであり，動作を学習，認知，生成することができる．

これを用い，Sugita and Tani (2005) 及び Ogata et al. (2007) はパラメータを共有する MTRNN と RNN を用い，ロボット動作と単語列をそれぞれ学習することで，相互想起を実現するモデルを提案した．また，Stramandinoli et al. (2012) は既に獲得している低次元における動作と言葉の対応関係に基づいて，RNN を用いることでより高次元の動作と言語表現の統合を目指した．

しかし，いずれの場合も動作パターンと固定の単語を組み合わせた表現のパターンとの対応関係学習である．本章では，より人間の言葉に対応するために，パターン学習ではなく，言葉の曖昧さを許容した，曖昧な言葉の意味理解に基づくロボットの動作生成を行う．

## 4.3 ロボット動作

本研究は家庭内でのロボット利用を目指しているため，複雑な動作が可能で，人に似せて作られているヒューマノイドロボットを用いる．次節より，本研究で用いるヒューマノイドロボット HIRONXC についての基礎知識と，動作構成，動作生成について説明する．

### 4.3.1 ヒューマノイドロボット HIRONXC

HIRONXC は（株）川田工業社製ヒューマノイドロボットである．HIRONXC はオープンなソフトウェア OpenHRP3 を持つプラットフォームで，双碗，双眼を持ち，左右手元にカメラがついている．Jython で動作する．

#### シミュレータ

（株）川田工業より提供されているシミュレータを図 4.1 に示す．Ubuntu12.04 で動作し，物体の設置，物体の認識，ロボットの動作が実機と同等に行うことができる．これを用い，動作を生成し安全を確認した後，実機で動作を確認する．

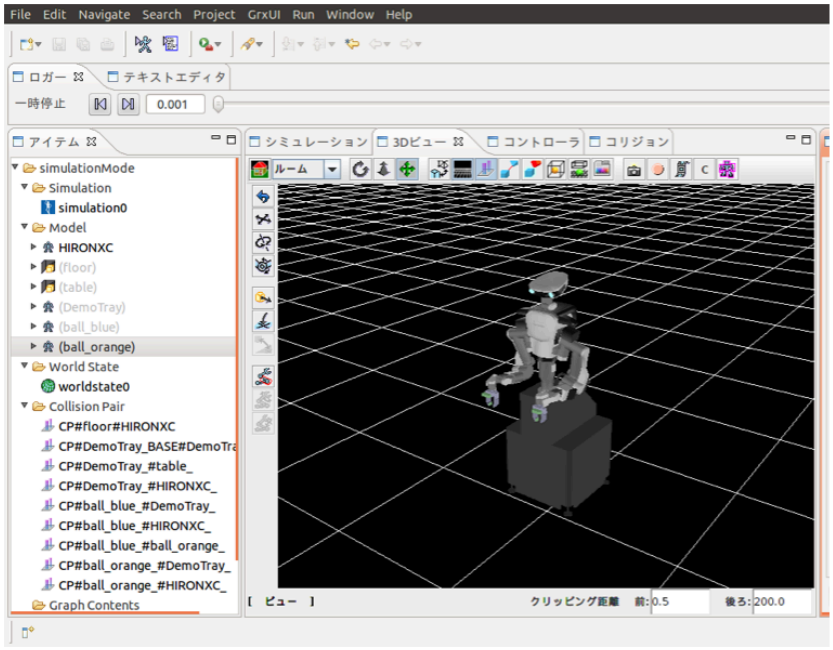



図 4.1: シミュレータ

### HIRONXC の動き

図 4.2[1]に HIRONXC の概観を示す.

HIRONXC は全部で 23 つの関節を持っており，体全体を動かす 1 つ（腰ヨー軸:CY），首の左右上下運動に 2 つ（首ヨー軸:NY，首ピッチ軸:NP），両腕に各 6 つ（右肩ヨー軸:RSY，右肩ピッチ軸:RSP．右肘ピッチ軸:REP，右手首ヨー軸:RWY，右手首ピッチ軸:RWP，右手首ロール軸:RWR，左肩ヨー軸:LSY，左肩ピッチ軸:LSP，左肘ピッチ軸:LEP，左手首ヨー軸:LWY，左手首ピッチ軸:LWP，左手首ロール軸 :LWR），両手に各 4 つ（右手関節:RH1, RH2, RH3,RH4，左手関節:LH1, LH2,LH3,LH4）の関節を持つ．それぞれの関節角と時間 $T$ を指定することで，$T$ 秒かけて指定された角度へと関節を動かすことが可能である．表 4.1 にそれぞれの関節が動く範囲を示す．ほぼ全ての関節で，1 秒で任意の角度に移動可能なことがわかる．

Jython を使い，HIRONXC を連続で動作させる場合，1 動作につき以下のフォーマットでの入力が必須であり，全ての関節角情報を HIRONXC に対して与えなければならない．

```
[[CY, NY, NP],[LSY, LSP, LEP, LWY, LWP, LWR],
 [RSY, RSP, REP, RWY, RWP, RWR],
 [RH1, RH2, RH3,RH4], [LH1, LH2,LH3,LH4],T]
```

[1]http://robot-support.kawada.jp/support/hiro/ より引用

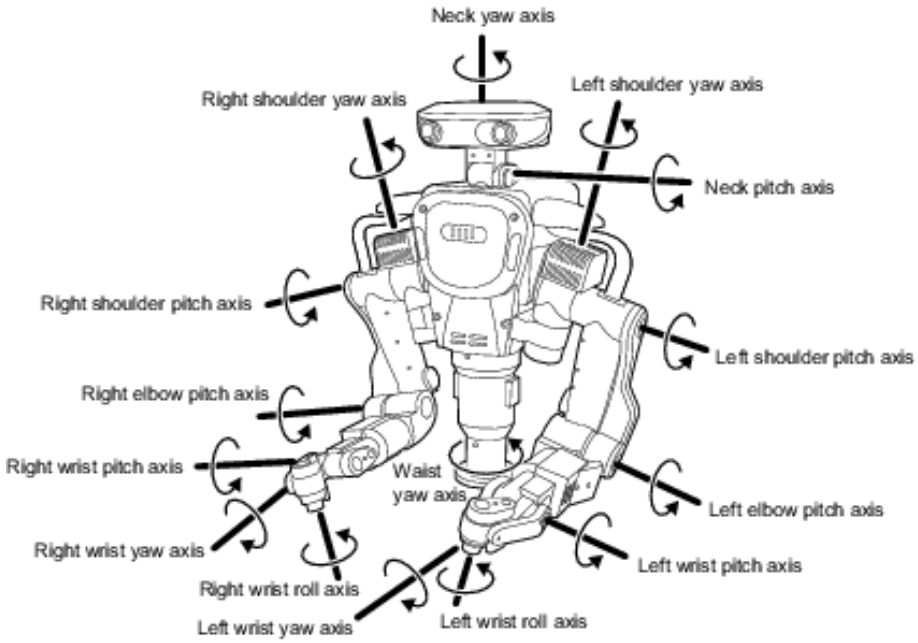



図 4.2: ロボットの関節

表 4.1: 関節の動作可能範囲

| 関節軸名称 | 記号 | 可動範囲 (deg) | 最大速度 (deg/s) |
|---|---|---|---|
| 腰ヨー軸 | CY | -163 to +163 | 130 |
| 首ヨー軸 | NY | -70 to +70 | 150 |
| 首ピッチ軸 | NP | -20 to +70 | 300 |
| 右肩ヨー軸 | RSY | -88 to +88 | 172 |
| 右肩ピッチ軸 | RSP | -140 to +60 | 133 |
| 右肘ピッチ軸 | REP | -158 to +0 | 229 |
| 右手首ヨー軸 | RWY | -165 to +105 | 300 |
| 右手首ピッチ軸 | RWP | -100 to +100 | 223 |
| 右手首ロール軸 | RWR | -163 to +163 | 300 |
| 左肩ヨー軸 | LSY | -88 to +88 | 172 |
| 左肩ピッチ軸 | LSP | -140 to +60 | 133 |
| 左肘ピッチ軸 | LEP | -158 to +0 | 229 |
| 左手首ヨー軸 | LWY | -105 to +165 | 300 |
| 左手首ピッチ軸 | LWP | -100 to +100 | 223 |
| 左手首ロール軸 | LWR | -163 to +163 | 300 |

### 4.3.2 動作構成

関節情報は可読性が低く，また人手でロボット動作を作成する際にも不便である．インバースキネマティクス（逆運動学）を利用することで，全ての関節情報を指示することなく，ロボットの特定の部位（例えば右手の先など）を目的位置に動作させることも可能であるが，本研究では複雑な動作への拡張を考え，ロボットを基本動作からより複雑な動作を構成するために，Cheng et al. (2013) による Activity-Attribute Matrix を参考にする．

#### Activity - Attribute Matrix

Activity-Attribute Matrix(以下 AAM) は動作と動作に関連している意味属性を符号化したものである．具体例を図 4.3 で示す．

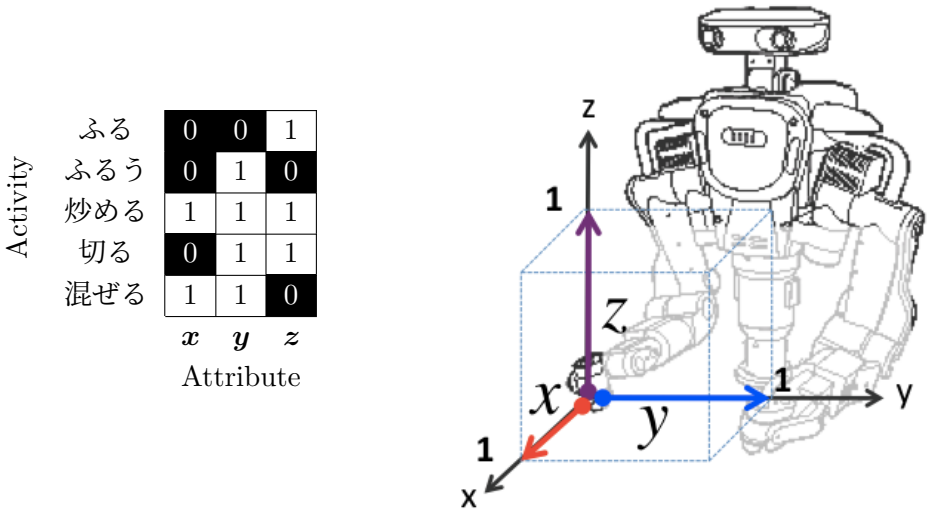



図 4.3: HIRONXC を用いた AAM 作成

$M$ を Activity（活動），$N$ を Attribute（属性）とし，各要素 $a_{ij}(i \in M, j \in N)$ において Attribute の Activity への含有関係について Activity $i$ を構成するのに Attribute $j$ が用いられている場合は 1，用いられていない場合 Attribute $j$ は 0 とする，$M \times N$ 行列により動作を表現する．図 4.3 では，Activity として調理動作を例に捉え，それに伴い Attribute として図 4.3 の右図ように右手の指先を前後に動かす ($\boldsymbol{x}$)，左右に動かす ($\boldsymbol{y}$)，上下に動かす ($\boldsymbol{z}$) を基本ベクトルを設定した．例えば，ふる動作について，$\boldsymbol{x} = 0, \boldsymbol{y} = 0, \boldsymbol{z} = 1$ となっているので，上下に動かす動きであることが分かり，また，切る動きは $\boldsymbol{x} = 0, \boldsymbol{y} = 1, \boldsymbol{z} = 1$ より，左右と上下の動きが含まれていることが分かる．

**時系列対応 AAM**

ロボットを実際に動かすにはそれぞれの関節角を指定する必要がある．そこで, Attribute の $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ それぞれに対して, 係数となるベクトルとの内積をとることにより, ロボットの動作生成を可能にする．さらに Activity を生成する過程において, それぞれの Attribute の度合い, 時系列性, 速度等が重要となるため, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ の変動割合にその動作を行うスピード $p_s = [0, 10]$ を組み合わせた $[p_x, p_y, p_z, p_s]$ を時系列に $n$ 個並べた $[[p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}], [p_{x_2}, p_{y_2}, p_{z_2}, p_{s_2}], \cdots, [p_{x_n}, p_{y_n}, p_{z_n}, p_{s_n}]]$ を与えることにより動作の生成を可能にする．表 4.2 に本研究で提案する時系列対応 AAM を示す．次章にて表 4.2 がどのように使用されるかについて具体例と共に示す.

### 4.3.3 動作生成

ロボットの動作生成の手順を以下に示す.

1. 基本ベクトルの準備
   要素動作となる $x, y, z$ 方向の基本ベクトルを準備する
2. 係数ベクトルの準備
   1. で準備したベクトル方向にどの程度移動させるかを示すベクトル（係数ベクトル）を生成する
3. 基本ベクトルと係数ベクトルの内積
   1. の基本ベクトルと 2. で生成された係数ベクトルの内積を計算し，関節角座標（以下,「座標」と呼ぶ）を求める
4. 移動先座標の取得
   3. で計算された座標に現在の座標を加えて，移動先の座標を取得する．動作時間 $T$ を指定し，ロボットに与える座標のフォーマットは以下の様になる.

```
[[CY, NY, NP],[LSY, LSP, LEP, LWY, LWP, LWR],
 [RSY, RSP, REP, RWY, RWP, RWR],
 [RH1, RH2, RH3,RH4], [LH1, LH2,LH3,LH4], T]
```

なお時間 T は，表関節の動作最大速度より，多くの関節でどの角度にも 1 秒以内で動作可能なため，$T_n = 10 - p_{s_n}$ とする．具体例を上記の順に従って以下に示す.

1. 基本ベクトルの準備
   準備される基本ベクトルは，表 4.2 の $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ に示される値となる.

表 4.2: 時系列対応 AAM の概要

| | $p_1$ | | | | $p_2$ | | | | ⋯ | $p_n$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 速く切る | 0 | 0 | -5 | 8 | 0 | 3.5 | 5 | 8 | | 0 | 3.5 | 5 | 8 |
| 速く混ぜる | 0 | 3.5 | 0 | 8 | 3.5 | -3.5 | 0 | 8 | | -3.5 | 0 | 0 | 8 |
| ゆっくり切る | 0 | 0 | -5 | 3 | 0 | 3.5 | 5 | 3 | | 0 | 3.5 | 5 | 3 |
| 細かく切る | 0 | 0 | -5 | 5 | 0 | 1 | 5 | 5 | | 0 | 1 | 5 | 5 |
| ザクザク混ぜる | 0 | 6 | 0 | 8 | 6 | -6 | 0 | 8 | | -6 | 0 | 0 | 8 |
| | $p_{x_1}$ | $p_{y_1}$ | $p_{z_1}$ | $p_{s_1}$ | $p_{x_2}$ | $p_{y_2}$ | $p_{z_2}$ | $p_{s_2}$ | ⋯ | $p_{x_n}$ | $p_{y_n}$ | $p_{z_n}$ | $p_{s_n}$ |
| | ↑ | ↑ | ↑ | | ↑ | ↑ | ↑ | | | ↑ | ↑ | ↑ | |
| | $\boldsymbol{x}$ | $\boldsymbol{y}$ | $\boldsymbol{z}$ | | $\boldsymbol{x}$ | $\boldsymbol{y}$ | $\boldsymbol{z}$ | | | $\boldsymbol{x}$ | $\boldsymbol{y}$ | $\boldsymbol{z}$ | |
| CY | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| NY | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| NP | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RSY | 0.1 | 1.8 | 0.7 | | 0.1 | 1.8 | 0.7 | | | 0.1 | 1.8 | 0.7 | |
| RSP | -2.3 | 0.7 | 0.1 | | -2.3 | 0.7 | 0.1 | | | -2.3 | 0.7 | 0.1 | |
| REP | 2.1 | -0.5 | -2.7 | | 2.1 | -0.5 | -2.7 | | | 2.1 | -0.5 | -2.7 | |
| RWY | 0.0 | 0.0 | 0.1 | | 0.0 | 0.0 | 0.1 | | | 0.0 | 0.0 | 0.1 | |
| RWP | 0.2 | 0.2 | 2.7 | | 0.2 | 0.2 | 2.7 | | | 0.2 | 0.2 | 2.7 | |
| RWR | 0.0 | -1.8 | 0.0 | | 0.0 | -1.8 | 0.0 | | | 0.0 | -1.8 | 0.0 | |
| LSY | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LSP | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | ⋯ | 0.0 | 0.0 | 0.0 | |
| LEP | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LWY | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LWP | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LWR | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RH1 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RH2 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RH3 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| RH4 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LH1 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LH2 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LH3 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |
| LH4 | 0.0 | 0.0 | 0.0 | | 0.0 | 0.0 | 0.0 | | | 0.0 | 0.0 | 0.0 | |

2. **係数ベクトルの生成**
   係数行列の生成を調理動作「切る」を例に説明する．

   切る動作は常に $\boldsymbol{x}$ が 0 で（前後の動きはなく），まず $\boldsymbol{z}$ が負（下へ）の動きをし，次に $\boldsymbol{z}$ が正（上へ）の動きをしながら $\boldsymbol{y}$ が正（左へ）動く，という特徴から生成される．このとき，動作の違いは特に $\boldsymbol{y}$（左右の動き）と時間 $t$ によって種類を生成できる．表 4.2 を見ると，複数の種類の「切る」という動作が，$\boldsymbol{y}$ と $t$ の値によって差別化されていることがわかる．

   次に，「速く切る」という動作を見てみると，$[[p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}], [p_{x_2}, p_{y_2}, p_{z_2}, p_{s_2}], \cdots, [p_{x_n}, p_{y_n}, p_{z_n}, p_{s_n}]]$ の値は [[0,0,-5,8],[0,3.5,5,8],⋯,[0,3.5,5,8]] となっており，時間の経過と共にその動作を示す係数ベクトルが表示されていることがわかる．

3. **基本ベクトルと係数ベクトルの内積，移動先座標の取得および速度の追加**
   表 4.2 中,「速く切る」における最初の係数ベクトルは，$[p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}] = [0, 0, -5, 8]$ であり，系数ベクトル $[p_{x_1}, p_{y_1}, p_{z_1}]$ と基本ベクトルの内積結果は

   ```
   [[0.0, 0.0, 0.0],
   [-3.5, -0.5, 13.5, -0.5, -13.5, 0.0],
   [ 0.6, 0, -100, -15.2, 9.4, -3.2],
   [0.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 0.0]]
   ```

   となる．現在の座標，つまり初期状態の座標

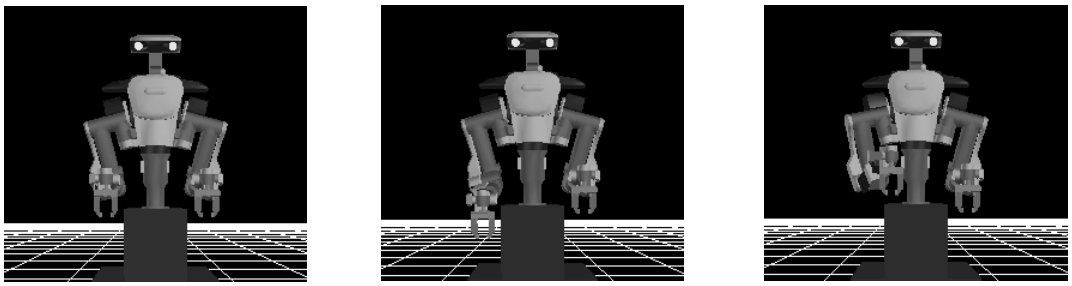
   ```
   Initialpose = [[0.0, 0.0, 0.0],
   [-0.6, 0, -100, 15.2, 9.4, 3.2],
   [ 0.6, 0, -100, -15.2, 9.4, -3.2],
   [0.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 0.0]]
   ```

   を内積結果に加え，さらに動作のスピード $p_{s_1}$ から計算した動作時間 $T_1 = 10 - p_{s_1}$ を組み合わせ，以下のようになる．

   ```
   [[[0.0, 0.0, 0.0],
   [-4.1, -0.5, -86.5, 14.7, -4.1, 3.2],
   [0.6, 0.1, -100, -15.2, 9.4, -3.2],
   [0.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 0.0]], 2]
   ```

   この動作生成を引き続き $[p_{x_2}, p_{y_2}, p_{z_2}, p_{s_2}] = [0, 3.5, 5, 8]$ にも適用した結果のロボットの動作を図 4.4 に示す．

初期状態と $p_1$ 終了時を比べると，ロボットの右手が下に降りており，$p_1$ 終了時と次の時刻の動作 $p_2$ 終了時を比べると，ロボットの右手が左斜め上にあがっており，想定した切るの動作がロボットで生成できていることが確認出来る．



初期状態　　$p_1$ 終了時　　$p_2$ 終了時

図 4.4: 「速く切る」の動作例

## 4.4 言語表現からの動作生成

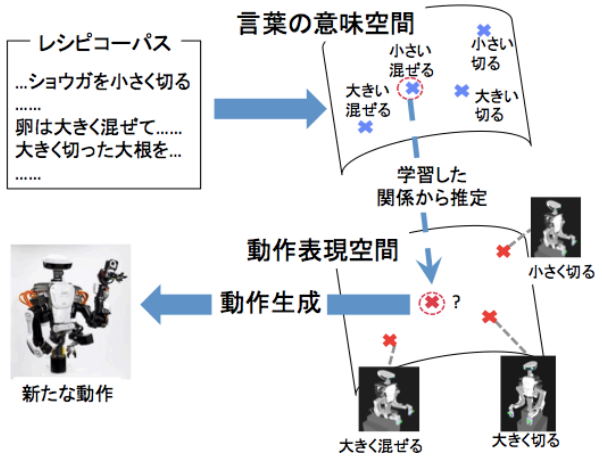



図 4.5: 提案手法の概要 (「小さく 混ぜる」の動作推定例)

動作を変化させる曖昧表現と動作を意味する動作表現とを組み合わせた言葉と，その言葉が意味する動作の対応関係が既知であるとする．動作との対応関係が分からない未知の言葉が与えられた時に，既知の言葉との意味関係から対応する動作を生成する手法を提案する．図 4.5 に提案手法の概要を例と共に示す．言葉は言葉の意味空間に，動作は動作表現空間にそれぞれ配置され，言葉と動作の関係により 2 つの空間の関係性を学習することで，言葉から動作を推定し，動作が生成できるようになる．例えば，「大きく 混ぜる」,「大きく 切る」,「小さく 切る」の 3 つの言葉とそれぞれの動作が既知であるとする．この時，未知の「小さく 混ぜる」動作を生成しようとする場合，既知の「大きく 切る」,「小さく 切る」の関係から「大きい」と「小さい」の関係性を推定でき,「大きく 混ぜる」動作と「大きい」と「小さい」の関係性を用いて「小さく 混ぜる」動作を推定し，動作を生成する．

## 4.5 ロボット動作制御モデルの構築

本節では, 曖昧表現と動作表現の組み合わせによって, 生成される動作が変化するモデルを 4 種類構築し，そのモデルの比較を行う．モデルの構造は重要である．Gaier and Ha (2019) は, ニューラルネットの重みを固定したまま, ニューラルネットの構造を変更することで, 複数のタスクである程度の精度を得た．自然言語処理分野では，Collobert and Weston (2008) が様々な問題に適用可能な共通の言語モデルの構築を深層学習を用いた Multitask Learning によって目指した．また，Kingma et al. (2014) は一部ラベル付けされた MNIST や SVHN のデータを用い，Variational AutoEncoder (VAE) (Kingma and Welling, 2013) によって自動に入力の特徴を捉え，文字を特徴に合わせて出力するモデルを構築した．本研究ではこれら先行研究を参考にモデルを構築する．また，入力をラベルとする場合と word2vec による分散表現とする場合についての比較を行い，分散表現の有用性についての検証も行う．なお，本研究では入力の検証とモデルの検証であるため，出力である動作は単純な動作とした．

### 4.5.1 実験仕様

言葉と動作の関係を学習するために，ニューラルネットワーク (NN) を用いる．図 4.6 にネットワークの概要を示す．ネットワークの構造は以下の 4 種類を比較する．

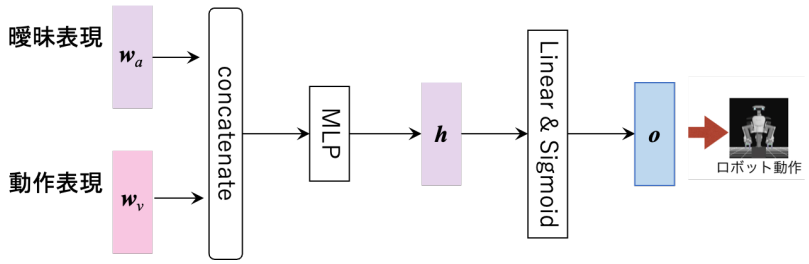
**Net1**： 曖昧表現 $\boldsymbol{w}_a$ と動作表現 $\boldsymbol{w}_v$ を結合して一つのベクトルとして入力し，ロボット動作表現 $\boldsymbol{o}$ を出す 3 層のニューラルネットワーク．中間層での活性化関数はシグモイド関数を用いる．

**Net2**： 曖昧表現 $\boldsymbol{w}_a$ と動作表現 $\boldsymbol{w}_v$ からそれぞれ二つの中間層 $\boldsymbol{h}_{a1}$,$\boldsymbol{h}_{a2}$ および $\boldsymbol{h}_{v1}$,$\boldsymbol{h}_{v2}$ を経た後結合し，ロボット動作表現 $\boldsymbol{o}$ を出すネットワーク．

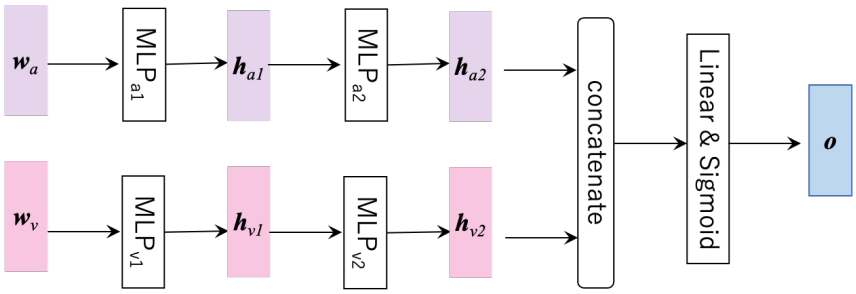
**Net3**： 曖昧表現 $\boldsymbol{w}_a$ と動作表現 $\boldsymbol{w}_v$ のそれぞれの中間層 $\boldsymbol{h}_a$ および $\boldsymbol{h}_v$ を結合し，中間層 $\boldsymbol{h}$ を経た後ロボット動作表現 $\boldsymbol{o}$ を出すネットワーク．

**Net4**： 動作表現によって曖昧表現が変化すると仮定する．まず，曖昧表現 $\boldsymbol{w}_a$ と動作表現 $\boldsymbol{w}_v$ のそれぞれの中間層 $\boldsymbol{h}_a$ および $\boldsymbol{h}_v$ を足し合わせ，動作表現により変化した曖昧表現の中間層 $\boldsymbol{h}_1$ と，動作表現の中間層 $\boldsymbol{h}_v$ を足し合わせ，中間層 $\boldsymbol{h}_2$ を経た後ロボット動作表現 $\boldsymbol{o}$ を出すネットワーク．

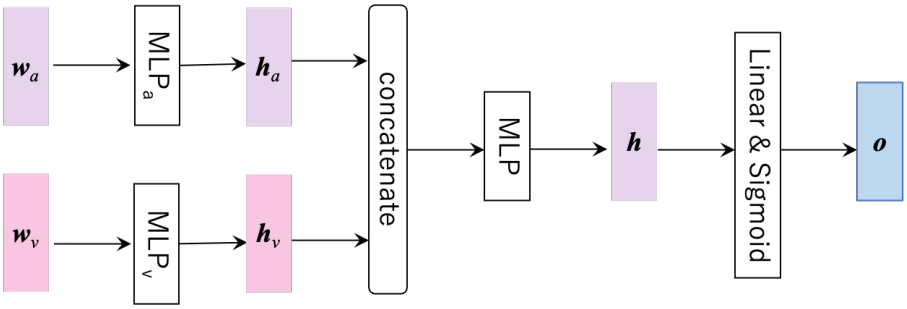
実際にロボットに動作をさせる場合には，動作表現 $\boldsymbol{o}$ を 4.3.3 項の手順により変換し，指示を行う．入力の曖昧表現 $\boldsymbol{w}_a$ および動作表現 $\boldsymbol{w}_v$ は，ラベル，つまり One-hot のベクトルとする場合と word2vec で作った分散意味表現とする場合で比較する．なお，Net1 の中間層での活性関数はシグモイド関数，Net2，Net3，Net4 の中間層での活性化関数はソフトプラ



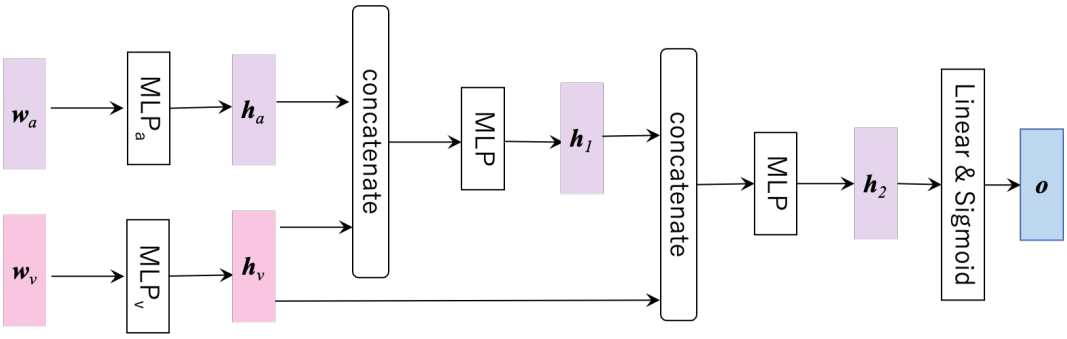

Net1




Net2




Net3




Net4

図 4.6: Network の概要

ス関数とする．ネットワークの学習方法は誤差逆伝播法を用い，またロボット動作の出力では活性化関数にシグモイド関数を用いた．言葉は，クックパッドのレシピに出現する調理動作表現で最も多い「切る」と 2 番目に多い「混ぜる」に係り受けしている曖昧表現のうち，動作の大きさや速さを表す曖昧表現 12 個 (「さっと」「ざっと」「ザクザク」「たっぷり」「手早い」「一気に」「少し」「細かい」「ちょっと」「しっかり」「きちんと」「じっくり」) と，その曖昧表現と組み合わせて使える調理に関する動作表現 6 個（「ふる」「ふるう」「炒める」「切る」「混ぜる」「つぶす」）を対象とする．入力の曖昧表現 $\boldsymbol{w}_a$ および動作表現 $\boldsymbol{w}_v$ をラベルとする場合は，曖昧表現 $\boldsymbol{w}_a$ は 12 次元，動作表現 $\boldsymbol{w}_v$ は 6 次元で表す．分散表現の場合は，クックパッドのレシピをクックパッドデータの材料コーパスを辞書として MeCab で分かち書きしたのち，word2vec の skip-gram を用いてベクトル化し，1 単語を 100 次元の分散表現で表す．曖昧表現 12 個 (「さっと」「ざっと」「ザクザク」「たっぷり」「手早い」「一気に」「少し」「細かい」「ちょっと」「しっかり」「きちんと」「じっくり」) の word2vec ベクトルの主成分分析結果の可視化を図 4.7 に示す．図 4.7 より，速度が速いことを表す「手

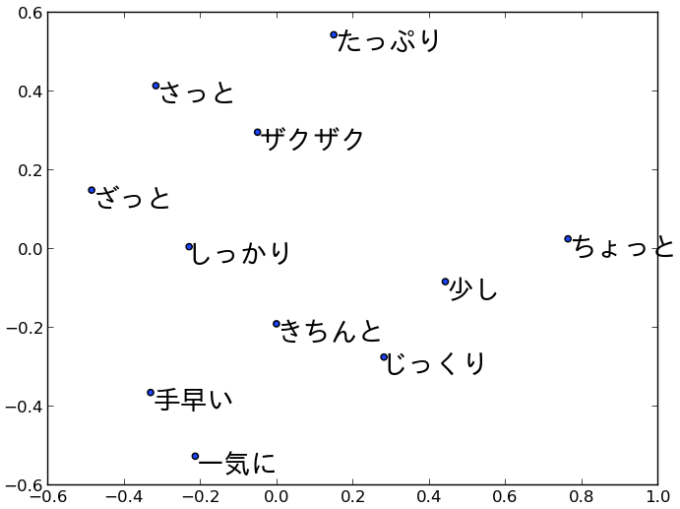



図 4.7: word2vec による曖昧表現の分散表現

早い」と「一気に」，少量であることを表す「少し」と「ちょっと」がそれぞれ近いことがわかる．一方で，逆の意味である「細かい」と「ザクザク」が近い．これは，word2vec による分散表現の作成前提である，分布仮説の「同じ文脈に現れる単語は似た意味を持つ傾向にある」により，対義語同士もまた同じ文脈に出やすいため，類義語と対義語は区別することができないことが原因である．類義語と対義語が近くなる問題を解消するために，潜在的意味解析をベースとした辞書知識を利用する手法や，意味的に関連している単語群は似たベクトルになるようにファインチューニングする手法があるが，本研究は分散表現の精度をあげることが目的ではないため，使用しない．

動作は，4.3.2 節に示した時系列対応 AAM を用いて人手で作成し，動作表現 6 個について表 4.3 に示すような $p_n = [p_{x_n}, p_{y_n}, p_{z_n}, p_{s_n}]$ の 4 次元からなる動作を用い，$p_1$ から $p_6$ の

表 4.3: 動作の時系列対応 AAM

| | $p_1$ | | | | $p_2$ | | | | $p_3$ | | | | $p_4$ | | | | $p_5$ | | | | $p_6$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 動作 | $p_{x_1}$ | $p_{y_1}$ | $p_{z_1}$ | $p_{s_1}$ | $p_{x_2}$ | $p_{y_2}$ | $p_{z_2}$ | $p_{s_2}$ | $p_{x_3}$ | $p_{y_3}$ | $p_{z_3}$ | $p_{s_3}$ | $p_{x_4}$ | $p_{y_4}$ | $p_{z_4}$ | $p_{s_4}$ | $p_{x_5}$ | $p_{y_5}$ | $p_{z_5}$ | $p_{s_5}$ | $p_{x_6}$ | $p_{y_6}$ | $p_{z_6}$ | $p_{s_6}$ |
| ふる | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 |
| ふるう | 0 | 3.5 | 0 | 5 | 0 | -3.5 | 0 | 5 | 0 | 3.5 | 0 | 5 | 0 | -3.5 | 0 | 5 | 0 | 3.5 | 0 | 5 | 0 | -3.5 | 0 | 5 |
| 炒める | 3.5 | 0 | 0 | 5 | -3.5 | 0 | 3 | 5 | 0 | 0 | -3 | 5 | 3.5 | 0 | 0 | 5 | -3.5 | 0 | 3 | 5 | 0 | 0 | -3 | 5 |
| 切る | 0 | 0 | -5 | 5 | 0 | 3.5 | 5 | 5 | 0 | 0 | -5 | 5 | 0 | 3.5 | 5 | 5 | 0 | 0 | -5 | 5 | 0 | 3.5 | 5 | 5 |
| 混ぜる | 0 | 3.5 | 0 | 5 | 3.5 | -3.5 | 0 | 5 | -3.5 | 0 | 0 | 5 | 0 | 3.5 | 0 | 5 | 3.5 | -3.5 | 0 | 5 | -3.5 | 0 | 0 | 5 |
| つぶす | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 | 0 | 0 | -3.5 | 5 | 0 | 0 | 3.5 | 5 |

連続する 6 つの動作により，24 次元のベクトルで表した．表 4.3 で示した動作のうち「切る」と「混ぜる」は図 4.8 のように表現出来る．

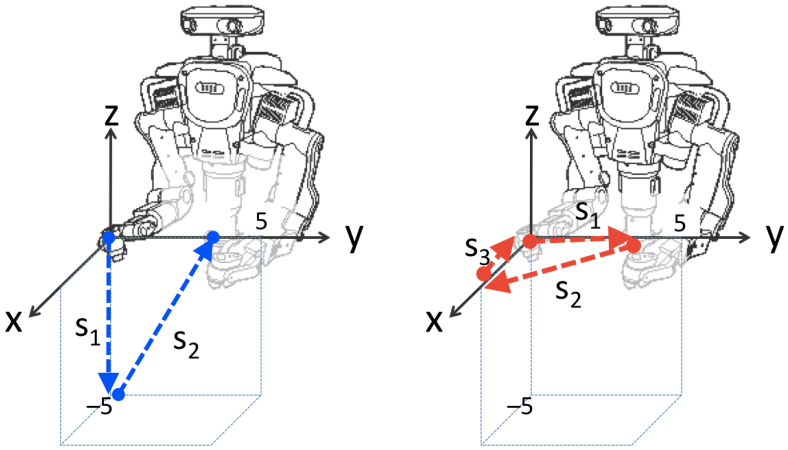



図 4.8: ロボット動作（左：切る，右：混ぜる）

「切る」の動きを例にとると，$p_1$ の動きは，$p_1$=$[p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}] = [0, 0, -5, 5]$ 次の $p_2$ の動きは，$p_2$=$[p_{x_2}, p_{y_2}, p_{z_2}, p_{s_2}] = [0, 3.5, 5, 5]$ で表されており，確かに図 4.8 左でも $p_1$ で $\boldsymbol{z}$ へ，$p_2$ で $\boldsymbol{y}$ と $\boldsymbol{z}$ への動きを同時に行っている．曖昧表現による動作の程度変化は，例えば「さっと」という動きは「速」くて「大きい」動作の程度であると仮定し，表 4.3 の着色部や速さを表す $p_s$ を大きくすることで表現する．他の曖昧表現も図 4.9 のように程度変化を仮定し，それぞれ表 4.3 の着色部や $p_s$ を変更することで動作を変化させる．なお，動作によって変化する $p_x, p_y, p_z$ は異なる．また，$p_x, p_y, p_z$ の大きさは [-10, 10] の範囲に限定し作成する．

以上を用いて, ネットワークを構築する．曖昧表現の次元数はラベルの場合は 12, word2vec の場合は 100, 動作表現の次元数はラベルの場合は 6, word2vec の場合は 100 とし，入力とする．出力層のノード数を $p_1 = [p_{x_1}, p_{y_1}, p_{z_1}, p_{s_1}]$ から $p_6 = [p_{x_6}, p_{y_6}, p_{z_6}, p_{s_6}]$ の 24 とした．中間層のノード数は 2 から 30 を試行し，入力を word2vec にした場合に最も平均二乗誤差

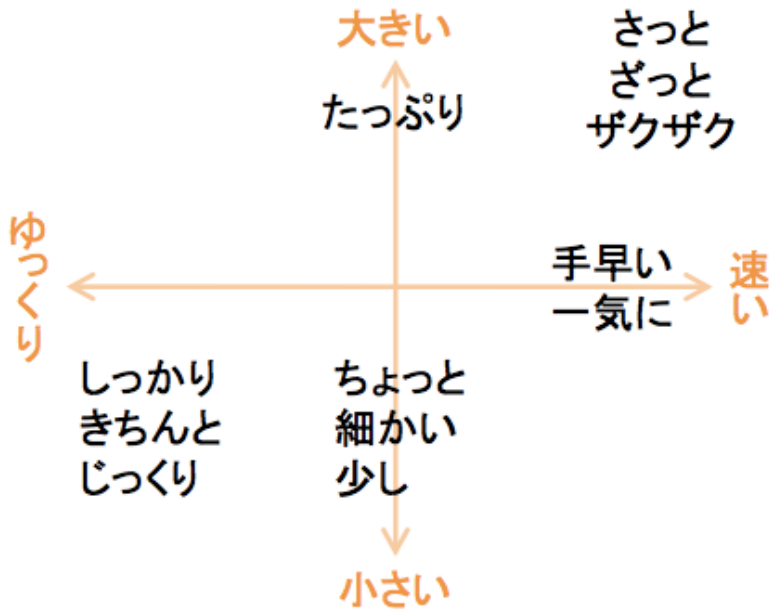



図 4.9: 曖昧表現の動作の程度変化

が小さくなった 20 とする[2]．訓練データとして，6 個の動作表現と，12 個の曖昧表現のうちいずれか 7 個をそれぞれ組み合わせた全 42 種の言葉に対し，各 100 個の動作を仮定した動作程度変化になるように切断正規分布により生成し，全 4200 個の動作を作成した．学習をした後，訓練データとして与えた曖昧表現と動作表現の組み合わせの 42 種，訓練データにない動作表現と曖昧表現の組み合わせの 30 種の動作を評価した．

### 4.5.2 評価指標

正解動作表現とネットワークにより出力された動作表現の生成結果の距離の差および時間の差を用いた誤差率によって，生成結果を評価する．実際にロボットで動作を行うには，生成結果を 4.3.3 項の手順により変換し，指示を行う必要があるが，物理的に動かすと誤差が生じるため，評価には生成結果を直接用いる．$p_1$ から $p_6$ の各時間終了時の正解動作と生成結果の距離の差 (単位：cm) を正解動作の移動距離で割った割合と，その動作にかかる時間の正解動作と生成結果の誤差 (単位：秒) を正解動作の時間で割った割合を足し合わせてその平均を取る．

$$\text{誤差率} = \frac{1}{12}\sum_{i=1}^{6}\left(\frac{\sqrt{(X_i - x_i)^2 + (Y_i - y_i)^2 + (Z_i - z_i)^2}}{\sqrt{X_i^2 + Y_i^2 + Z_i^2}} + \frac{|T_i - t_i|}{T_i}\right) \tag{4.1}$$

[2]入力をラベルとした場合，中間層のノード数が 9 で最も平均二乗誤差が小さくなったが，ノード数 20 との差は僅かであったため，ノード数 20 を用いる

ここで $X_i$，$Y_i$，$Z_i$，$T_i$ は時刻 $p_i$ の正解動作，$x_i$，$y_i$，$z_i$，$t_i$ は生成結果である．この誤差率が 0 に近いほど正解動作と似た動きになっており，より正しく生成できていることを表している．

### 4.5.3 実験結果

表 4.4 から表 4.6 は式 (4.1 の誤差率を算出し，パーセントで換算した評価結果である．

表 4.4 は入力の言葉をラベル，つまり One-hot にした場合と word2vec による分散表現にした場合のネットワーク毎の生成動作誤差率の平均を表している．ここで,「学習済み表現」とは入力の曖昧表現と動作表現の組み合わせに対応する動作が学習データに含まれている場合の生成動作の誤差率平均,「未学習表現」とは，入力の曖昧表現と動作表現を組み合わせた動作が学習データに含まれていない場合の生成動作の誤差率平均である．

表 4.4: ネットワーク比較

| | | 誤差率平均 | |
|---|---|---|---|
| 構造 | 入力処理 | 学習済み表現 | 未学習表現 |
| Net1 | ラベル | 42.05 | 47.49 |
| | word2vec | 40.74 | 47.70 |
| Net2 | ラベル | 32.13 | 47.78 |
| | word2vec | 32.08 | 46.91 |
| Net3 | ラベル | 8.79 | 14.00 |
| | word2vec | 8.44 | 12.71 |
| Net4 | ラベル | 8.93 | 12.54 |
| | word2vec | 8.48 | 12.10 |

表 4.4 より，学習済み表現を入力とする場合でも未学習表現を入力とする場合でも，Net1 の未学習表現の入力を除いて word2vec を用いた場合の方がラベルを用いた場合よりも誤差率が若干小さくなった．しかしその差は大きくなく，入力処理よりも，ネットワーク構造が重要であることを示していると言える．

ネットワーク構成について，Net1 と比較し，Net2，Net3，Net4 では誤差率が小さくなった．よって，入力の方法として曖昧表現と動作表現を連結した場合より，分けて入力とした場合が有用である．また Net3，Net4 が Net2 よりも誤差率の平均は小さくなっており，動作表現と曖昧表現の言葉を足し合わせた後，ロボット動作を出力する前に一度中間層を経たネットワーク構成の学習が有用であると言える．また Net4 はの未学習表現の入力に対

する誤差率の平均が最小であり，未学習表現の入力に対しては，動作表現によって曖昧表現が変化するという仮定を用いると有効であると言える．

表 4.5，表 4.6 は入力を word2vec のベクトルとした場合の Net2 と Net4 の生成動作の誤差率を表している．この時，着色部は曖昧表現と動作表現の組み合わせによる動作が学習データに含まれない，未学習表現の入力を与えた場合であり，それ以外（白色部）は学習データに含まれている曖昧表現と動作表現の組み合わせを入力とする場合の誤差となっている．表 4.5 と表 4.6 を比較すると，Net2 の表 4.5 は，動作表現毎の誤差率の平均が 14.5%から 50.6%となっており，ばらつきがある一方，Net4 の表 4.6 では動作表現毎の誤差率の平均が 5.0%から 12.9%と大きくは変わらないことから，Net4 では各動作の特徴を捉えることができていると言える．また，１つの曖昧表現における動作表現毎の誤差率について，Net2 の表 4.5 は，白色部と着色部にばらつきがある一方，Net4 の表 4.6 は白色部と着色部に大きな差が見られないことより，曖昧表現が動作毎に変化する学習できていると言える．

表 4.5: Net2 評価結果

| | | 動作表現 | | | | | |
|---|---|---|---|---|---|---|---|
| | | ふる | ふるう | 炒める | 切る | 混ぜる | つぶす |
| 曖昧表現 | さっと | 30.1 | 30.6 | 26.9 | 18.4 | 29.4 | 31.4 |
| | ざっと | 31.8 | 31.3 | 28.0 | 19.7 | 30.7 | 34.9 |
| | ザクザク | 28.2 | 36.7 | 26.4 | 20.3 | 36.6 | 41.5 |
| | たっぷり | 21.7 | 23.1 | 18.4 | 10.2 | 24.9 | 25.8 |
| | 手早い | 9.7 | 12.7 | 12.4 | 9.1 | 13.8 | 15.9 |
| | 一気に | 8.6 | 10.7 | 13.2 | 8.7 | 12.3 | 11.4 |
| | 少し | 99.3 | 140.2 | 65.6 | 16.1 | 83.6 | 145.7 |
| | 細かい | 79.1 | 52.8 | 25.3 | 13.7 | 48.3 | 43.1 |
| | ちょっと | 39.0 | 31.6 | 14.9 | 9.1 | 27.2 | 26.2 |
| | しっかり | 158.0 | 130.3 | 50.6 | 17.7 | 92.4 | 84.2 |
| | きちんと | 54.4 | 61.4 | 38.8 | 18.3 | 58.8 | 73.5 |
| | じっくり | 46.9 | 29.3 | 15.8 | 13.0 | 31.4 | 23.8 |
| 平均 | | 50.6 | 49.2 | 28.0 | 14.5 | 40.8 | 46.5 |

表 4.6: Net4 評価結果

| | | 動作表現 | | | | | |
|---|---|---|---|---|---|---|---|
| | | ふる | ふるう | 炒める | 切る | 混ぜる | つぶす |
| 曖昧表現 | さっと | 9.0 | 9.0 | 9.4 | 8.8 | 10.1 | 10.1 |
| | ざっと | 11.3 | 7.9 | 10.6 | 7.5 | 9.0 | 12.7 |
| | ザクザク | 6.1 | 7.5 | 8.9 | 6.5 | 10.4 | 11.2 |
| | たっぷり | 0.9 | 1.2 | 1.7 | 0.8 | 1.6 | 2.5 |
| | 手早い | 8.4 | 10.0 | 9.0 | 9.8 | 10.0 | 10.6 |
| | 一気に | 6.8 | 8.5 | 7.4 | 7.5 | 9.7 | 7.5 |
| | 少し | 18.8 | 37.0 | 9.3 | 1.7 | 13.7 | 28.8 |
| | 細かい | 15.1 | 14.7 | 6.1 | 1.2 | 12.0 | 7.1 |
| | ちょっと | 6.7 | 7.9 | 4.6 | 1.7 | 2.9 | 4.4 |
| | しっかり | 25.9 | 23.1 | 12.3 | 4.8 | 19.9 | 18.2 |
| | きちんと | 14.6 | 17.2 | 14.8 | 6.5 | 22.4 | 15.8 |
| | じっくり | 7.1 | 10.2 | 8.2 | 2.9 | 11.9 | 9.7 |
| 平均 | | 10.9 | 12.9 | 8.5 | 5.0 | 11.1 | 11.6 |

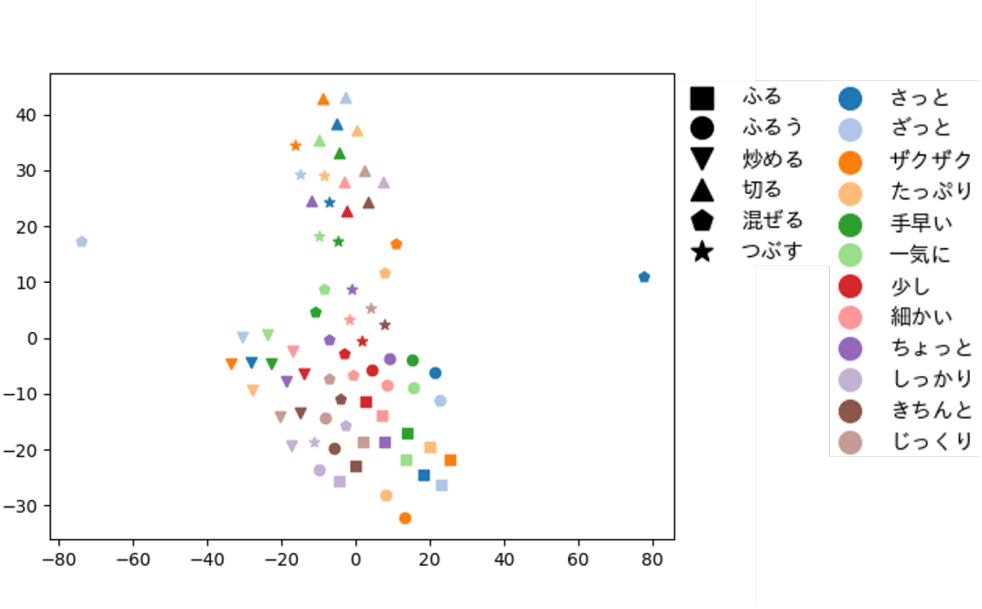



図 4.10: 正解動作表現 $\boldsymbol{o}$ の可視化

高次元での距離分布が次元削減してもできるだけ合致するように変換が行える t-SNE を用いて，正解動作表現を次元削減し，可視化した結果を図 4.10 に，Net2 と Net4 の中間層

と出力層を次元削減し可視化した結果をそれぞれ図 4.11 および図 4.12 に示す．それぞれ，色によって入力の曖昧表現を区別し，マークの形によって入力の動作表現を区別している．

図 4.10 の正解動作表現と Net2 および Net2 と Net4 を比べると，正解動作表現では，同一のマークは比較的近くに分布しており，また同一の色同士も近い傾向にある．同一の動作表現による動作表現同士は近く，また同一の動作表現による関係性には劣るが，同一の曖昧表現による動作表現同士も近い傾向にあると言える．Net4 の出力層も同様の傾向にあることがわかる．一方，Net2 の出力層は，同一のマークで固まっており，同一の色同士の関係性はみられない．このことから，曖昧表現を捉えられていないことが分かる．

中間層を比較すると，Net2,Net4 それぞれの最初の中間層 $\boldsymbol{h}_{a1}$,$\boldsymbol{h}_{v1}$ および $\boldsymbol{h}_{a}$,$\boldsymbol{h}_{v}$ では，同一のマーク同士，もしくは同一の色同士でまとまっている．入力に近い中間層であるため，より入力の曖昧表現や動作表現を反映した分布になっていると考えられる．次の層では，Net2 において $\boldsymbol{h}_{a2}$ および $\boldsymbol{h}_{v2}$ は，最初の層と比べて，より同一のマーク同士，もしくは同一の色同士が近くなっている．つまり，同一の曖昧表現や動作表現による中間層のベクトル同士がより近い分布となっている．正解動作は曖昧表現による動作表現の関係性より，動作表現による動作表現の関係性の方が強いため，より $\boldsymbol{h}_{v2}$ を反映した出力 $\boldsymbol{o}$ となったと考えられる．

一方 Net4 では $\boldsymbol{h}_1$，$\boldsymbol{h}_2$ とも同一のマークおよび同一の色の両方が近い傾向にある．つまり，曖昧表現および動作表現の両方の傾向を持つ中間層のベクトルとなっている．その結果，出力 $\boldsymbol{o}$ も曖昧表現および動作表現を反映できたと考えられる．

生成した動作表現をロボットに指示した場合に想定される動きの例を示す．「ザクザク」「混ぜる」の $p_1$ から $p_3$ までの動作は図 4.13 のようになる．Net1 に比べ Net2，Net3，Net4 の 3 つの手法は，より「混ぜる」動きができており，特に Net3，Net4 は予想データとより似た動きになることが確認出来る．

word2vec を入力として学習したネットワークは，学習データに存在しない表現（ここでは未知語と呼ぶこととする）を入力として与えることが可能である．例えば，動作表現の「切る」と共に，未知語の「ちゃんと」の word2vec のベクトルを入力に与えた場合と，学習データに存在する既知語の曖昧表現の「きちんと」を入力に与えた場合，および，未知語の「キャベツ」の word2vec のベクトルを入力に与えた場合と，既知語の曖昧表現の「ザクザク」を入力に与えた場合のそれぞれの出力動作を図 4.14 に示す．「ちゃんと」「切る」と「きちんと」「切る」,「キャベツ」「切る」と「ザクザク」「切る」はそれぞれ近い動きであるが，わずかに差異を持つ動作が生成できた．「ちゃんと」と「きちんと」,「キャベツ」と「ザクザク」は word2vec のベクトルがそれぞれ近い．言葉での関係性から動作の変化を推測し，動作生成できていると言える．

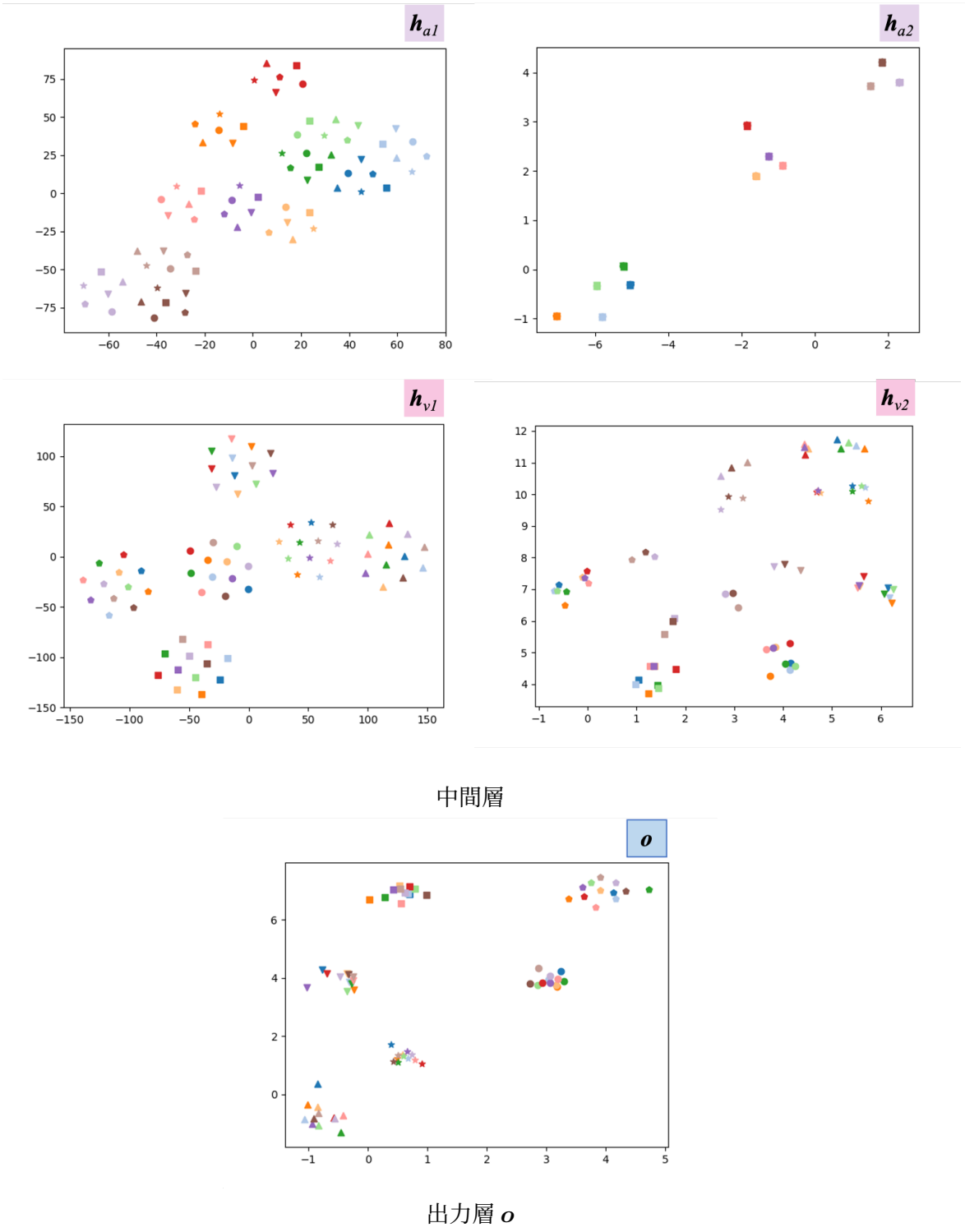
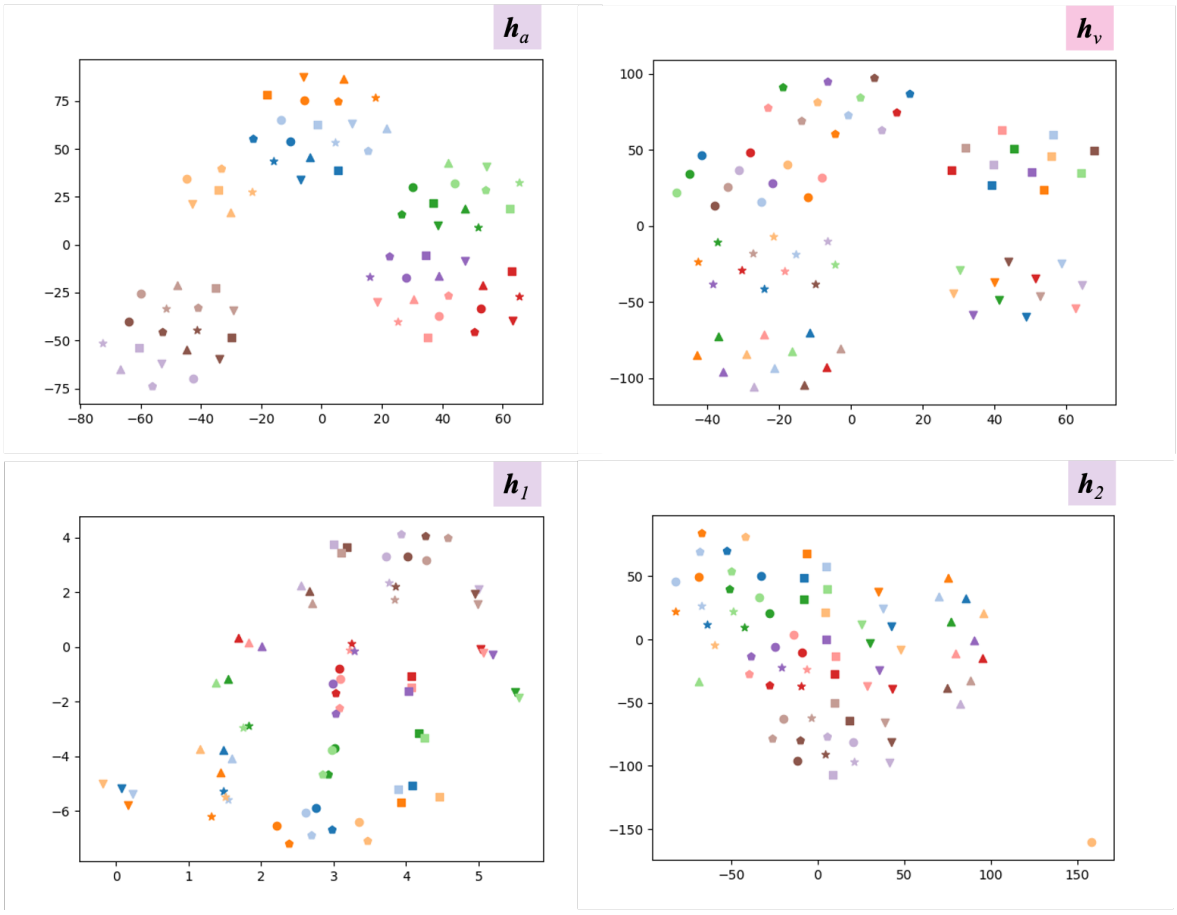
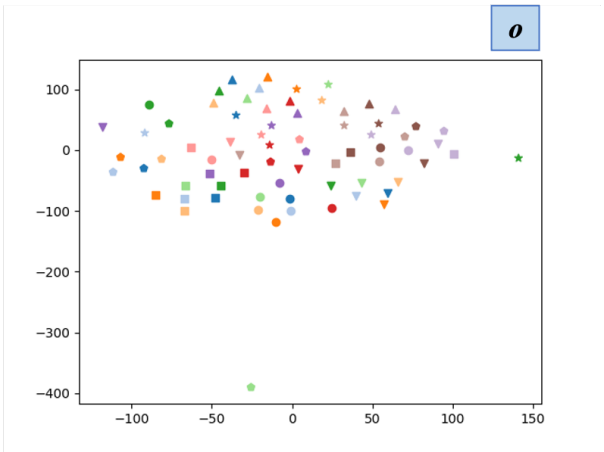



図 4.11: Net2 の中間層と出力動作表現の可視化

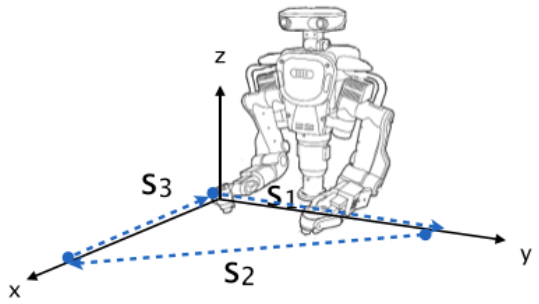



中間層

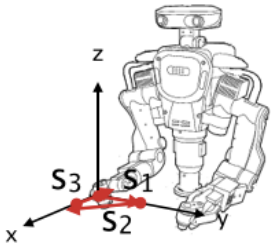



出力層 $o$

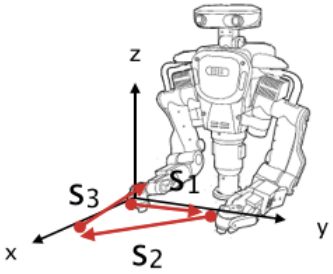
図 4.12: Net4 の中間層と動作表現の可視化



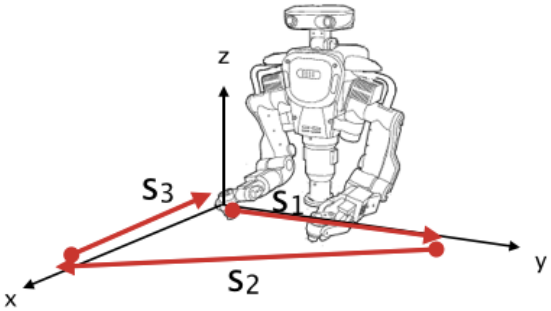

正解動作



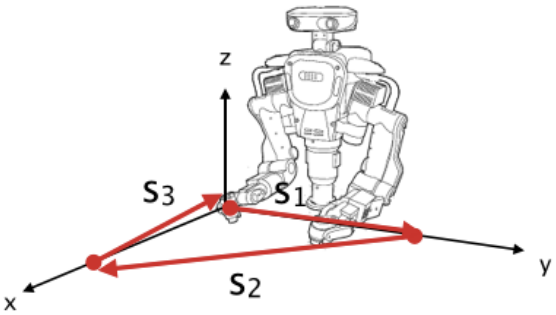

Net1




Net2

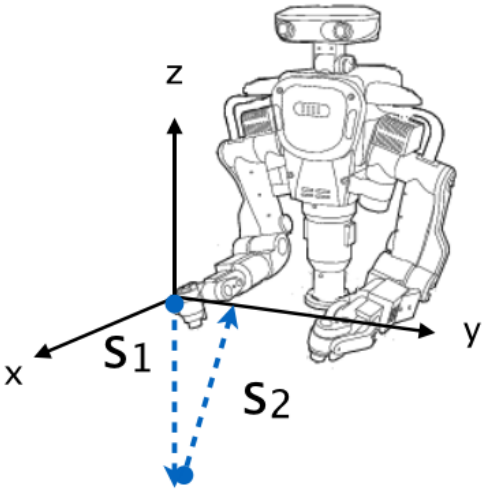



Net3

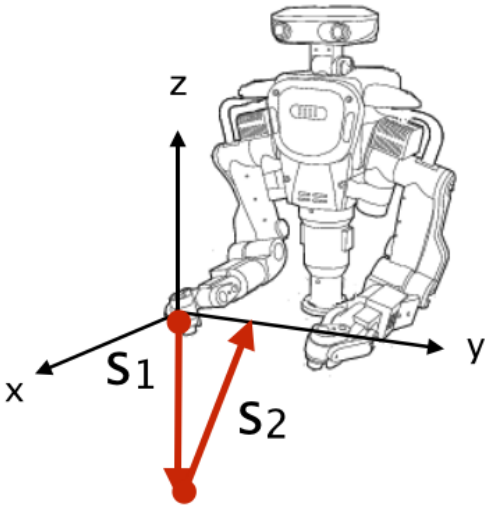



Net4

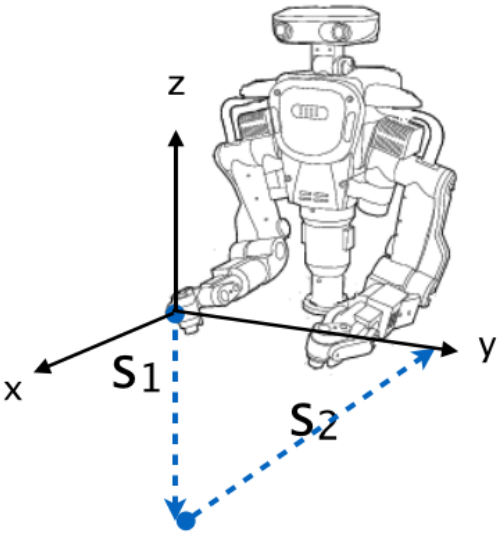
図 4.13: 「ザクザク」「混ぜる」のロボット動作



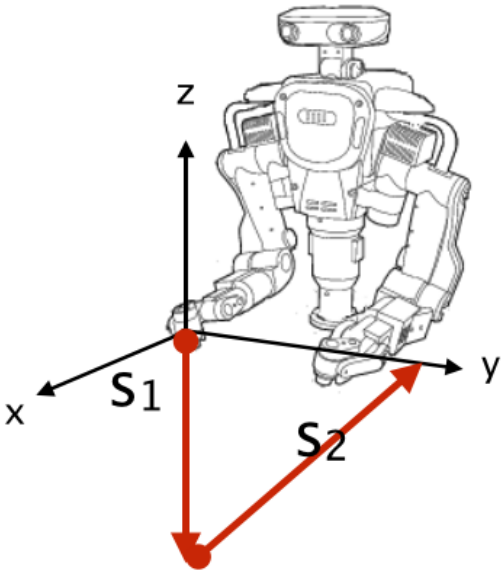

既知語：「きちんと」




未知語：「ちゃんと」




既知語：「ザクザク」




未知語：「キャベツ」

図 4.14: 未知語入力時の「切る」ロボット動作

## 4.6 まとめと今後の課題

本章では言葉と動作の対応関係学習によるロボット動作生成手法の提案，特に複数の動作生成に適したネットワークについて検証した．一部の言葉と動作の関係が既知の場合，ラベルを入力にした場合も　word2vec を入力とした場合も，曖昧表現と動作表現の組み合わせによる言葉と動作の関係性を学習していない場合にも動作推定可能であり，特に word2vec を用いた場合は言葉をラベルとして与えた場合に比べ誤差が小さくなった．一方，word2vec を用いたことによる精度向上よりも，言葉と動作の関係性獲得に用いたネットワークの構成による精度の差が大きくなった．入力層について，曖昧表現と動作表現を連結した場合に比べそれぞれ入力を行った方が良い結果となった．また，適切な中間層の形を得，さらに動作表現により曖昧表現が変化するという仮説が示された．なお入力に word2vec を用いたネットワークについて，全く行ったことのない言語表現の動作を生成することができ，既知の表現からの推測ができていると考えられる．以上により，調理で使われがちな曖昧な表現と動作表現を組み合わせた言葉による指示で，ロボットを動かすことができ，家庭内でロボットへの指示に慣れていないユーザーの使用に適したロボット動作手法を獲得した．本枠組みにより，言葉の曖昧さを動作というモダリティに限定して対応関係を獲得することで，モダリティ変換することができた．

今後の課題として，肘を上げたまま動かす，動きを連続させるなどのより複雑なロボット動作への適用，「にんじんを小さく切る」，「ごぼうを小さく切る」で大きさが変わるように，動作対象に対して動作が変わると考えられるため 3 語以上の入力への対応や，「きれいに」や「ちゃんと」のような，より曖昧な言葉の入力に対しての動作生成が挙げられる．また，家庭内で使う場合には口頭での指示が考えられ，非文法文や音声認識誤りが想定されるが，その対応として，過去の動作からの推定や，周りの状況を判断できるような仕組みの導入が必要と考えられる．なお本課題では word2vec を用いているため，レシピコーパスに出現していない単語に対して，ゼロ頻度問題が起こりうるが，例えば subword の組み合わせによって，対応できると考えられる．

# 第5章　一般ドメイン実況生成

本章では，非言語から言語を生成する課題として，一般ドメイン実況生成課題を取り上げる．一般ドメイン実況生成の概要を述べた後，関連研究を説明する．データセットの作成手順の説明，データセットの分析を述べ，課題設定について説明する．各課題に対して実験を行い，評価する．

## 5.1　概要

動画や画像に対して自然言語でその内容を記述する研究は広く行われている (Vinyals et al., 2015b; Xu et al., 2015; Venugopalan et al., 2015a; Gao et al., 2017; Pei et al., 2019; Wang et al., 2018)．動画に対する言語生成課題として代表的な課題として，動画キャプション生成 (Krishna et al., 2017; Zhou et al., 2018a) が挙げられる．これは，動画内の重要なイベントを複数検出し，それぞれに対して説明テキストを生成する課題である．一方，動画内で起きているイベントの客観的な説明や主観的な感想を，視聴者が動画の再生と共に音声として聴く，あるいは字幕テキストとして読むことができる実況は，視聴者が動画の状況をより深く理解でき，また動画視聴をより楽しむことができる．すなわち，実況を付与することにより動画の価値向上が期待できる (Schaffrath, 2003)．実況は，視聴者に映像の内容をわかりやすく伝えるために，映像とともに音声として出力されたり，映像にテキストを被せ提示されたりする．そのため，実況発話を開始するタイミングや発話の長さを適切に制御することが求められる．実況生成は既存研究である動画キャプション生成と類似するが，1) タイムスタンプの重なり，2) タイムスタンプの時間幅の観点から，動画キャプション生成とは異なる設定である．図 5.1 に，既存の動画キャプションデータセットである ActivityNet Captions (Krishna et al., 2017) のキャプション，および本研究で生成を目指す実況の例を示す．動画キャプション生成ではイベント検出をまず行い，それぞれに対しキャプションを生成するため，イベントの発生する時間幅が重なる場合がある．一方，実況では前の発話が終わってから次の発話を始める必要がある．さらに，動画キャプションではリアルタイムに発話することが想定されていないため，図 5.1 の 1 つ目のキャプションに示すように，短い時間幅に対し長いテキストが対応づくことがある．一方，実況では時間幅と発話の長さが関係する特徴がある．

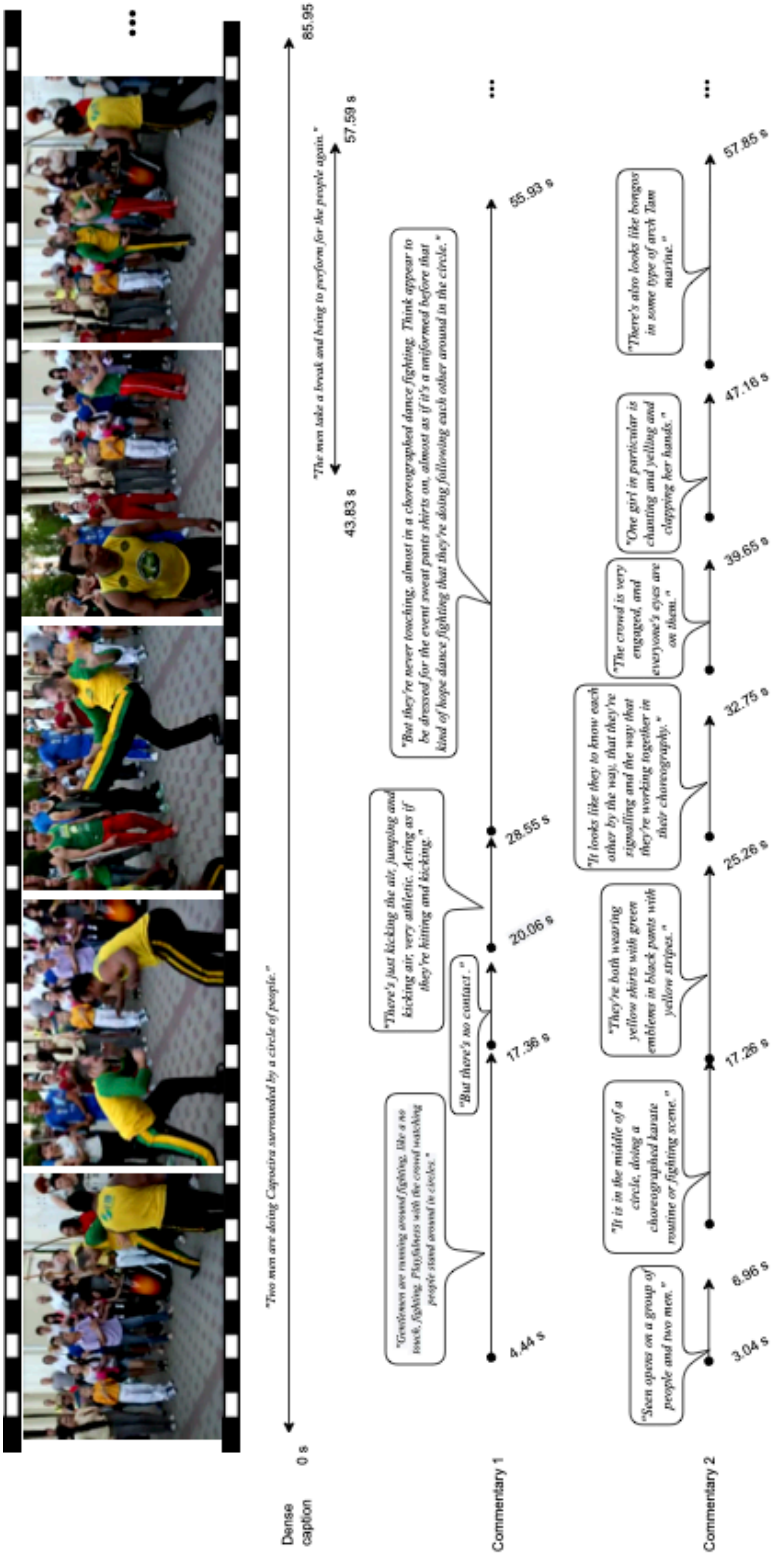



図 5.1: 既存の動画キャプションデータセットと提案の実況との比較.

このように実況は，時間的位置，内容，ビデオ内のオブジェクトの相関が，ビデオ内のイベントと部分的にしか整合しておらず，これまでの既存の画像キャプション生成課題とは明確に異なるものである．

一般的に実況は，スポーツ中継やゲームにおいて，観客を興奮させ，没入させる重要な役割を持っている (Schaffrath, 2003)．このような背景から，これまでにも解説を自動生成するアプローチが提案されているが，分野固有の情報を活用することが可能な特定分野を対象としたものが一般的であった．例えば，RoboCup サッカーシミュレーションと連動した実況生成 Mike (Tanaka-Ishii et al., 1998) では，試合の時々刻々の選手の位置やボール位置によって実況生成が行われ，レーシングゲーム実況生成 (Ishigaki et al., 2021) では，レーシングカーの速度やレースの進捗度といったレーシングゲーム特有の情報を用いて生成を行っている．このように従来の実況生成の研究では，分野固有の情報を活用することが可能な特定分野，つまりクローズドメインを対象としたものに限られていた．そこで本章では，より汎用性の高い技術を目指すため，特定ドメインでなく一般ドメインの動画を対象とする動画の実況生成タスクを提案する．具体的には，ダーツやサッカーなどのさまざまなスポーツ，縫い物や編み物の様子，手を洗うなど，人の動作の動画を集めた ActivityNet コーパスに含まれる動画に対して実況を生成するタスクとする．分野固有の情報を使用することなく，実況を生成する手法の獲得を目指し，これが達成されることで，例えばテレビ番組の字幕放送や投稿動画の実況付加，CM 作成など，ドメインが限定されない動画に対して説明文などを付与したい場合に実用できる技術となりうる．

まず，様々なドメインの人間の動作を含むビデオに対して実況を収録し，新しい大規模データセットの構築を行なった．また，実況生成課題に取り組むために，一般的なニューラルネットワークに基づく手法を提案する．提案する手法の長所と限界を理解するために，我々のデータに基づく詳細な実証研究も提示する．

## 5.2 関連研究

言葉を用いて動画内の状況を説明する機械学習手法の開発は，computer vision (CV) の分野と自然言語処理の分野の両方にまたがる挑戦である．オープンドメインの動画に対して，説明を付与する課題に取り組んだ最初の研究は Venugopalan et al. (2015b) である．この動画キャプション課題は，30 秒未満の短い動画に対して，主となる内容を自然言語で記述する課題であった．この課題は後に， Krishna et al. (2017) によって，ビデオで発生する複数のイベントを検出し，自然言語によってそれぞれのイベントを説明する課題に拡張された．このように，ビデオで発生する複数のイベントを検出し，説明文を生成する課題を Dense Video Captioning (DVC) と呼ぶ．DVC は一般的に，1. まず入力動画中のイベントが起きている箇所 *proposal* を特定し，2. その後，特定した *proposal* 区間に対して、キャプション生成モデルが説明テキストを生成する，のように，2 つのサブタスクに分けて課題解決を目指している．なお，最近では Transformer を用いた end-to-end アプローチも提案

されている (Zhou et al., 2018b; Wang et al., 2021).

Taniguchi et al. (2019) は，データからテキストに変換して生成する手法として，試合のイベントデータ（関与した選手，ピッチ内の位置，プレイの種類など構造化されたデータ）を用いて，ウェブページに表示するサッカーの試合の解説文を生成する手法を提案した．また最近では， Kim and Choi (2020) が野球の試合映像から，試合の合間に表示する解説を自動生成する手法を提案した．しかしこれらの手法は，動画と一緒に表示されることを想定していない解説を生成することを目的としているため，本論文で提案する課題とは異なる．またこれらの手法は，それぞれの課題を解決するためにドメイン固有のデータに依存している点も本課題とは大きく異なる点である．

Taniguchi et al. (2019) ではもちろんのこと，Kim and Choi (2020) では，解説文の生成はドメインオントロジーに依存するだけでなく，入力動画からゲームの詳細をモデル化することを目的とした，いくつかのサブコンポーネント（選手検出器や投球結果認識器など）に基づいている．さらに最近， Ishigaki et al. (2021) はレーシングカーゲームの解説を自動生成するタスクを提案し，ゲームプレイ動画と音声解説のアノテーションデータセットを初めて公開した． Ishigaki et al. (2021) は構造化されたデータ，具体的にはテレビゲームエンジンから直接抽出された車の位置と速度，ハンドルの角度を含むテレメトリデータを用い，実況文解説を生成するモデルに適用した．これらの特徴を利用することで，動画情報とテキストのみを入力とするモデルよりも大幅な性能向上を実現し，動画内容を説明する課題におけるドメイン内情報の重要性を改めて示した．

本章で行うタスクは，リアルタイムに入出力を行う点において，同時通訳 cho (2016) とも似ている（詳しくは，5.4 節を参照）．このような課題の場合，文を生成するために多くの情報を待つことと，より早く文章を生成することのトレードオフを考慮したモデルを提案する必要がある．

## 5.3 データセット

一般ドメインの動画実況データセット作成手順およびデータ分析について述べる．データ作成は実況音声の収集と文字起こしの 2 段階の手順で行った．

### 5.3.1 実況音声収集

クラウドソーシングサービスの Amazon Mechanical Turk (MTurk) を用い，人の動作の動画を集めた ActivityNet コーパス (Fabian Caba Heilbron and Niebles, 2015) に含まれる YouTube 動画のうち，2021 年 2 月の時点で YouTube 上に存在する動画を実況付与対象とした．また，動画は 30 秒以上 4 分以下のものに絞った．作業者には，図 5.2 に示す画面

（HIT）を表示し，動画を視聴しながら動画内で起こっていることを主観的な感想なども含め実況するよう指示した．作業者が録音を停止しない場合，動画終了後 1 分後に強制的に録音が停止されるよう設計した．作業者は 2 つの動画に対して二回録音を行い，1 動画につき 2 人以上の作業者が実況を録音した．1 回目の録音では動画をはじめて視聴する場合の実況，2 回目の実況では動画の内容をすでに把握している状況での実況を収集することを目指している．データセットに含める録音の品質を向上させるため，Pydub[1]を用いて音量調節および雑音削除した．また，音声録音時間が動画時間の 90%以上 200%以下のものに絞り，動画の長さと録音時間が極端に乖離する事例をフィルタリングした．さらに，沈黙時間が極端に長い低品質な録音を除外するために，発話している時間が動画の長さの 30%以上 110%未満であるものに限定した．

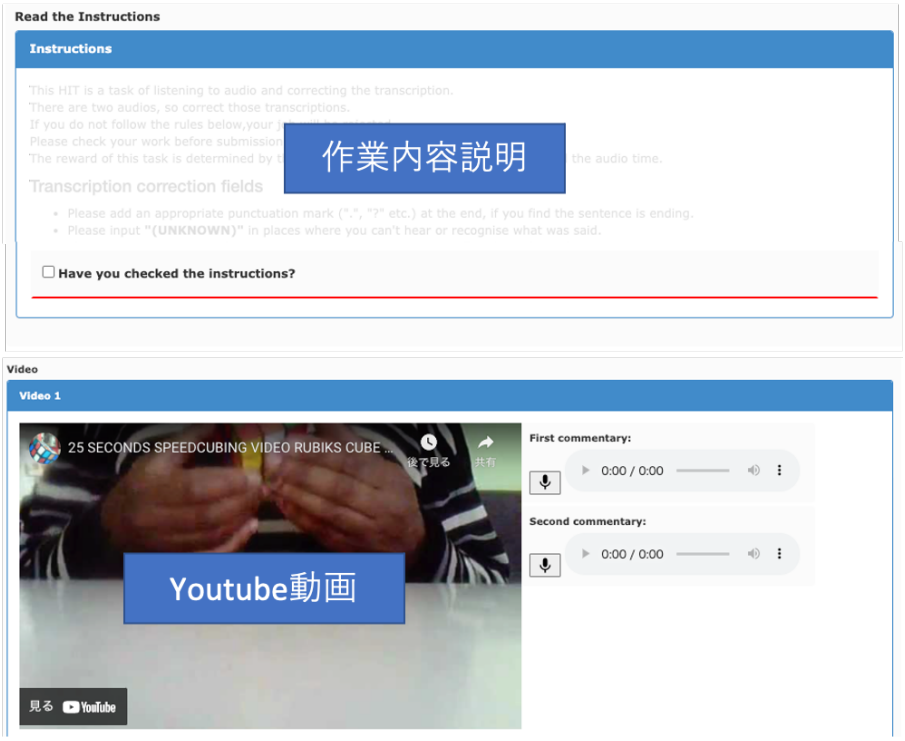



図 5.2: MTurk で用いた実況録音用 HIT の例

### 5.3.2 自動文字起こしと実況テキスト修正

次に収集した実況音声を AWS の音声のテキスト変換機能 Amazon Transcribe[2] を用いて自動文字起こした．自動文字起こしにより，発話中の各単語にタイムスタンプが付与され

[1]https://github.com/jiaaro/pydub
[2]https://aws.amazon.com/jp/transcribe/

表 5.1: 実況テキスト例

| 種類 | テキスト |
|---|---|
| **Amazon Transcribe** | *"uh Let's see uh there's a man, there's a man throwing darts and I think oh it seems one of them,he threw three darts, and I think 11 of them hit the center, didn't it?."* |
| 文字起こし修正 | *"uh Let's see uh there's a man, there's a man throwing darts and I think oh it seems one of them, he threw three darts, and I think one one of them hit the center, didn't it?."* |
| フィラー等削除 | *"There is a man throwing darts. He threw three darts. I think one of them hit the center."* |

る．自動文字起こしの結果には誤りが含まれる場合があり，また発話中にはフィラーや言い淀みが含まれるため，Mturk にて作業者に文字起こし修正と，フィラーや言い淀み，言い直しの削除や修正作業を依頼した．テキスト修正用の HIT の例を図 5.3 に示す．作業者には自動文字起こしされたテキストを発話単位に分割して表示し,，各発話に対して自動文字起こしの修正と，フィラー等の削除を指示した．さらに，実況音声と共に実況録音時に使用した動画を再生する UI を用いて，実況内容と動画との乖離がないかを合わせて確認できるよう設計した．表 5.1 に Amazon Transcribe を用いて音声をテキスト変換した結果，MTurk で文字起こしを修正した結果，フィラーなどを削除した結果の例を示す．作業者が提出した修正テキストのうち，数字をスペルアウトできているか，フィラーを削除できているかを基準として，最終的な文字起こし修正テキストおよびフィラー等削除済実況テキストを収集した．

### 5.3.3 データセット統計および既存データとの比較

作業者が意味のある実況を提出したことを確認するために，Krishna et al. (2017) で作業者が意味のあるイベントをアノテーションしているか確認した手法 (Baldassano et al., 2017) に従い，ある作業者の各発言タイムスタンプと他の作業者の発言タイムスタンプの重なりを調査した．その結果，temporal Intersection over Union (tIoU) が 45.5%となった．これは，Dense video captioning のデータセットである ActivityNet Captions (Krishna et al., 2017) の 59.5%の tIoU より若干低い[3]が，本課題であつかう実況における，人によって実況を行う時間帯が若干異なるという性質上，許容範囲と考えられる．また，ActivityNet Captions と本実況データの平均 tIoU は 18.7%となり，実況者がイベントをより明確にするために発言を遅らせたり，逆に急がせたりするために，同時刻の動画中のイベントと発言との間に強い整合性がないことを再度示唆していると考えられる．

[3]Krishna et al. (2017) の論文値では 70.2%だったが，再現できなかった

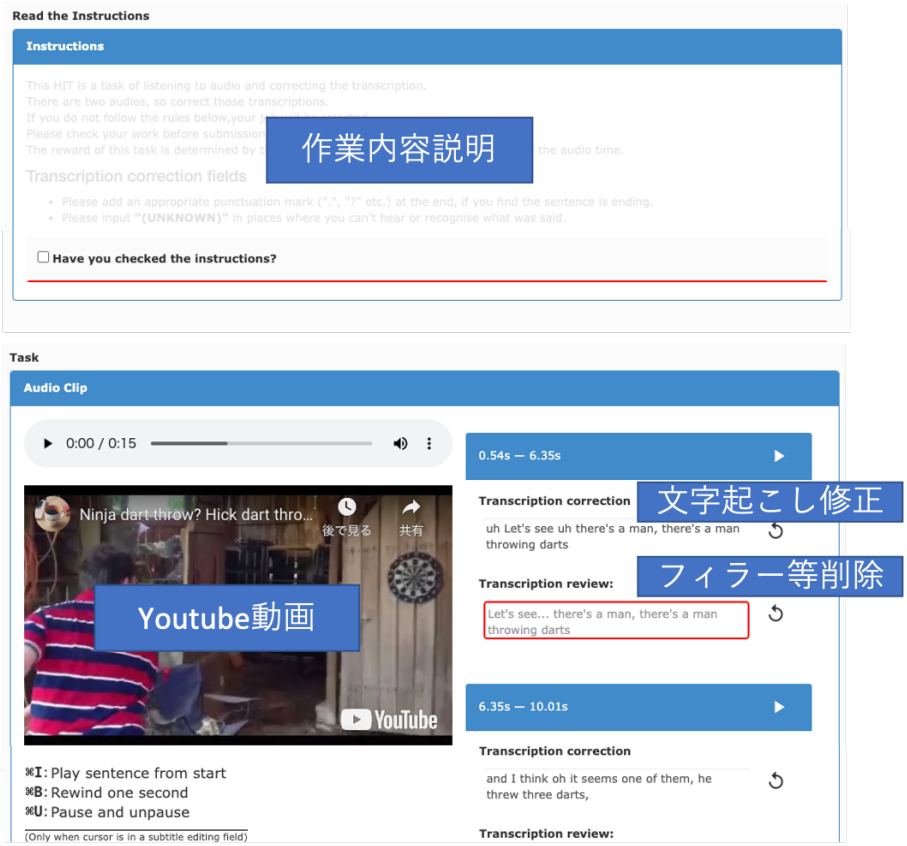



図 5.3: MTurk で用いた文字起こしテキスト修正 HIT の例

さらに，実況内容の一致を実況者間で調査するため，DVC のようなタスクを評価する時によく使われる，BLEU-4 とシーングラフの一致度合いを F 値で評価する SPICE (Anderson et al., 2016) の 2 種類を用いる．本データセットと ActivityNet Captions の 2 つのデータセットをそれぞれ用い，同一動画における異なる作業者のテキストを時間順に並べて比較する．ActivityNet Captions では、BLEU-4 と SPICE の平均値がそれぞれ 4.86 と 0.12 であったのに対し，本データセットではそれぞれ 2.59 と 0.034 であった．この結果は，実況は人によって内容が変わりうるという，本課題の性質を反映していると考えられる．図 5.4 と図 5.5 は，ランダムに選択した 2 つの動画（動画 ID が *“c1RR1cmS9LU”*，*“KU8VVtam3ig”* の動画）に対する本データセットと ActivityNet Captions の BLEU-4 と SPICE の内容一致評価結果をそれぞれ可視化したものである．ActivityNet Captions と比べて，本データセットでも異なるアノテーションに対して同程度の内容の一致を示していることが観察でき，この結果は本データセットの実況の品質をある程度担保していると考えられる．

作成したデータセットの統計を表 5.2 に示す．合計 715 人の作業者によって 6,771 本の動画に対して，合計 25,000 件以上の実況テキストを集めた．図 5.1 は実際に収取したデータで，2 人の実況者が動画に対して付与した発話の例を示しており， Krishna et al. (2017)

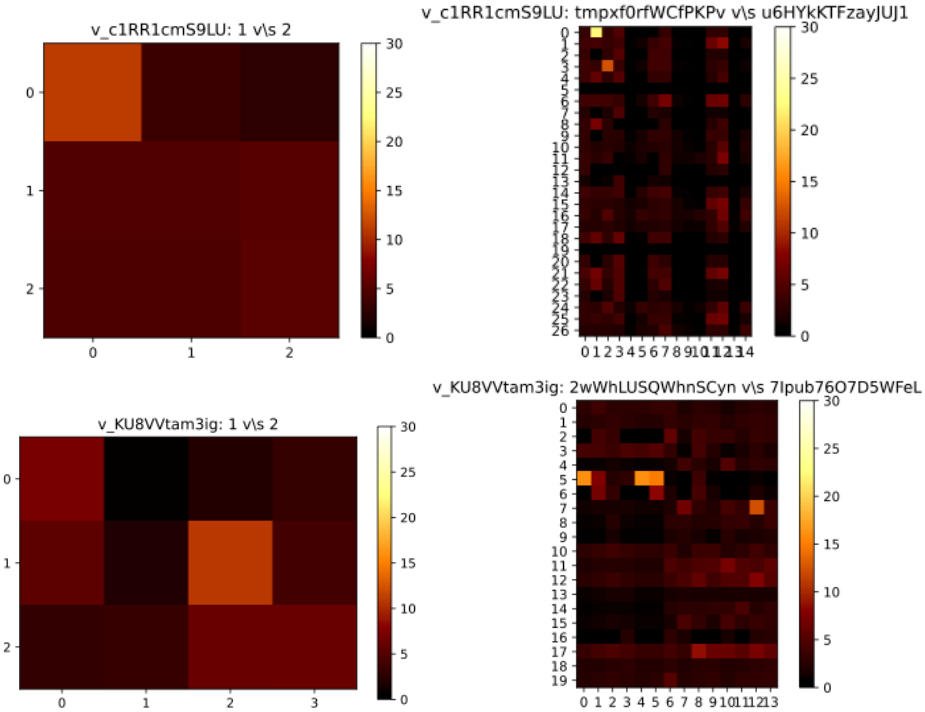



図 5.4: ActivityNet Captions(左) と本データセット (右) の BLEU-4 による内容一致評価可視化.

による同一動画の DVC と比較している．この例から分かるように，本データは動画上に密に分布しており，平均して動画時間の 78 ± 21%を占めている．本課題と最も近い Ishigaki et al. (2021) によるクローズドメインのデータセットでは 55 ± 17%であり，それと比べてもより密に分布している．また、1 動画あたりの平均発言数は 17.64 となり，Ishigaki et al. (2021) の 52.25 に比べ非常に少ない．本データセットでは発言間の無音時間も短いことから，発言数の少なさは本データセットで用いてた動画が短いためと推察される．

## 5.4 問題設定

本節では，一般ドメイン実況生成課題に向けたモデルの提案を行う．以降，与えられた動画 $V \in \mathcal{V}$ が $V = \{v_f\}_{f=1}^{F}$ のようなフレームのシーケンスとして特徴付けられると仮定する．また各動画には，開始と終了のタイムスタンプ $t_s$ と $t_e$ で定義される非重複区間において，ある期間に何が起こっているかを実況解説した自然言語発話シーケンス $U = \{u_t\}_{t=1}^{T}$ がアノテーションされている．

実況生成課題として，以下の 2 種類の設定が考えられる．

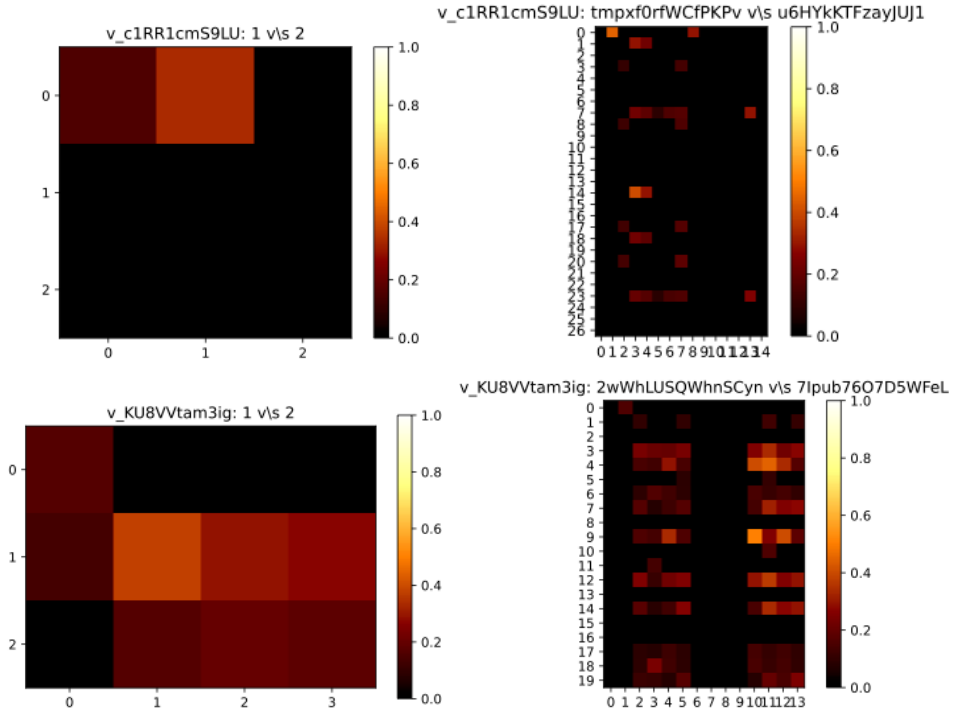



図 5.5: ActivityNet Captions(左) と本データセット (右) の SPICE による内容一致評価可視化.

- *Off-line Live Commentary Generation*: 録画された動画に対する解説の自動生成．学習時と推論時に動画全体にアクセスできる．
- *Real-time Live Commentary Generation*: ライブ映像に対する解説の自動生成，追加のデータを待つことと、長い遅延なしに発話を生成することのトレードオフのバランスをとる必要がある．同時通訳に類似したマルチモーダルな設定とみなすことができ，より挑戦的な設定である．

本論文では Off-line 設定に焦点を当てるが，本実証研究は 5.2 節で示すように，今後行うであろう Real-time 設定の課題に関係している．

関連する先行研究 (Taniguchi et al., 2019; Kim and Choi, 2020) やクローズドメインの実況生成 (Ishigaki et al., 2021) では大きなタスクをそのまま処理することなく，パイプラインアプローチで組み合わせることができる，より単純なサブタスクに分割する慣習がある．本研究でもそれを踏襲し，2 つのサブタスクに分割する．

- **タイミング推定:** 各発言のビデオ内の適切な開始と終了のタイムスタンプを識別するサブタスク．本研究で対象とするデータの一般ドメイン性を考慮すると，このタスク

表 5.2: 本データセットと (Krishna et al., 2017), (Ishigaki et al., 2021) の統計値比較.

| データセット | Krishna et al. | 本データセット | Ishigaki et al. |
|---|---|---|---|
| 課題 | DVC | 実況生成 | |
| ドメイン | 一般ドメイン | | クローズドメイン |
| 動画数 | 14,926 | 6,771 | 2,473 |
| - 動画時間合計 | 487:35:43 | 239:58:03 | 226:37:53 |
| - 最大動画時間 | 755 s | 238 s | 452 s |
| - 最小動画時間 | 1 s | 30 s | 246 s |
| アノテーション数 | 19,811 | 25,817 | 2,473 |
| - 作業者数 | - | 715 | 5 |
| - アノテーション時間の合計 | 742:23:29 | 904:01:24 | 226:37:53 |
| - 合計文章数 | 71,957 | 455,485 | 129,226 |
| - 1 動画あたりの平均文章数 | 3.63 | 17.64 | 52.25 |
| - アノテーション重複率 | 10% | 0 % | 0% |
| - 平均沈黙時間 | - | 0.84 s | 3.46 s |

を適切に解決するには，モデルは与えられた粒度の各タイムポイントにおいて，ビデオ内の関連するイベントを認識できる必要がある．

- **発話生成:** 入力された映像のセグメントに対して、適切な発話を生成するタスク．

以降，簡単のため，与えられた入力例 $V$ に焦点を当てる．すべての場合において，$F$ 個の入力動画フレームを動画特徴量のシーケンス $G = \{\boldsymbol{g}_n \in \mathbb{R}^{d_v}\}$ $(n = 1, \ldots, N)$ にマップする動画エンコーダ[4] と，タイムスタンプ $t$ を特徴量インデックス $\tau \in \{1, \ldots, N\}$ に変換できるマッピング関数が存在すると仮定する．

### 5.4.1 タイミング推定

これまでの実況生成におけるタイミング推定には，領域内の情報を広範囲に使用してきた．本タスクは一般ドメインであるため，正規化された知識ベース，もしくはタスクに有用な情報をすでに獲得している事前学習済みのモデルを利用して，外部の知識にアクセスすることが重要であると考えた．そこで，Temporal Action Localization (TAL) を提案する．TAL は CV の基本タスクで，トリミングされていない動画シーケンス中の人間の活動を時間的に位置づけることを目的とする．このタスクは従来から Dense Video Caption(DVC)

[4] このマッピングは $\tau = (t \cdot n \cdot \mathrm{fps})/N$ によって，フレーム/特徴量インデックスを時刻に変換する．

に不可欠な要素であり，モデルは ActivityNet (Fabian Caba Heilbron and Niebles, 2015) を利用して学習することが多い．一般に，モデルは “proposal”，すなわち，目的の動作が配置されうる時間の範囲を生成する．ここで，proposa は予測される開始時間と終了時間 $t_m^s$, $t_m^e$ の $p_m = (t_m^s, t_m^e, c_m)$ からなる $\mathbb{P} = \{p_m\}_{m=1}^M$ と表し，$c_m$ はモデルの確からしさである．

proposal $p_m$ に対して, タイムスタンプを特徴インデックス $\tau_m^s$ と $\tau_m^e$ に $1 \leq \tau_m^s < \tau_m^e \leq N$ で対応付けし，$p_m$ を $G$ を用いて $K = \tau_m^e - \tau_m^s + 1$ 個の特徴量 $\boldsymbol{g}_{\tau_m^s}, \ldots, \boldsymbol{g}_{\tau_m^e}$ と表す．まず，各 roposal に対する固定長表現 $\boldsymbol{p}_m$ を以下の式で求める：

$$\boldsymbol{x}_k = \mathrm{GRU}_E(\boldsymbol{g}_{\tau_m^s+k-1}, \boldsymbol{x}_{k-1}) \quad k = 1, \ldots, K, \tag{5.1}$$

この時，$\boldsymbol{x}_k$ は $\boldsymbol{x}_0 = 0$ を初期状態とする $\mathrm{GRU}_E$ の $k$ 番目の隠れ状態で，$\boldsymbol{p}_m = [\boldsymbol{x}_1; \boldsymbol{x}_K]$ はセグメント表現である．各 proposal の開始時刻に従ってセグメント表現を並べて，次に示すようにポインタネットの初期コンテキスト表現を生成する役割を持つ別の GRU に入力する：

$$\boldsymbol{h}_m = \mathrm{GRU}_C(\boldsymbol{p}_m, \boldsymbol{h}_{m-1}) \quad m = 1, \ldots, M. \tag{5.2}$$

$\mathrm{GRU}_C$ の初期隠れ状態を 0 ($\boldsymbol{h}_0 = 0$) とし，その最後の隠れ状態を用いて，$\boldsymbol{s}_0 = \boldsymbol{h}_M$ として Pointer Net を初期化する．次時刻以降の proposal の集合を 1 つずつ繰り返し，以下の操作を実行する：

$$score(\boldsymbol{s}_i, \boldsymbol{x}_m) = \boldsymbol{v}^T \tanh([\boldsymbol{s}_i; \boldsymbol{x}_m]) \tag{5.3}$$

$$a_{i,m} = \frac{\exp(score(\boldsymbol{s}_i, \boldsymbol{x}_m))}{\sum_{m=1}^M \exp(score(\boldsymbol{s}_i, \boldsymbol{x}_m))} \tag{5.4}$$

$$\boldsymbol{s}_i = \mathrm{GRU}_P([\boldsymbol{x}_{m^\star}; pos(m^\star)], \boldsymbol{s}_{i-1}) \tag{5.5}$$

ここで, $\boldsymbol{v}$ は学習可能なパラメータのベクトル, $\boldsymbol{a}_i$ は大きさ $M$ のベクトル, $pos(m)$ は proposal $p_m$ を大きさ $d_p$ のバイナリーベクトルで正規化する関数であり，$m^\star = \arg\max_m(a_{i,m})$ ($m^\star \in \{1, \ldots, M\}$) である．式 (5.5) に示す通り，各ステップにおいて proposal 集合に対する確率分布を計算し，最も可能性の高い proposal を選択する．次のステップの Pointer Net の隠れ状態を計算するため，選択された proposal 表現 $\boldsymbol{x}_{m^\star}$ をそれの位置表現と連結して $GRU_P$ に入力する．入力には特別な $EOS$ という proposal を $p_0$ として組み込み，モデルは全ての proposal が選択されるまで，あるいは Pointer Net が $p_0$ を選択するまで反復を繰り返す．モデルは，以下に定義する損失を最小化し，正解と高い重複を持つ proposal を選択するように学習される：

$$\mathcal{L} = -\sum_{r=1}^{R}\sum_{m=1}^{M} \mathrm{tIoU}(s_r, p_m) \log a_{r,m} + (1 - \mathrm{tIoU}(s_r, p_m)) \log(1 - a_{r,m}), \tag{5.6}$$

ここで，$\{s_r\}_{r=1}^R$ は正解 proposal の系列，$a_{r,m}$ は $r$ 番目の proposal が $m$ 回目の反復で選ばれる尤度，$\mathrm{tIoU}(\cdot,\cdot)$ は temporal Intersection over Union である．

### 5.4.2 発話生成

発話生成モデルには, いくつかのマルチモーダルなタスク (Chen et al., 2020; Hong et al., 2021) で有効性が示されている，Transformer モデル (Vaswani et al., 2017) を用いる．先行研究に従い, 動画およびテキストのコンテキスト両方をモデルに入力する．具体的には, 前の発話に由来するトークン列 $s_j$ と、各正解アノテーションの 3 秒前から終了までの $(t_s-3, t_e)$ の動画セグメント $v_t$ を入力する．これは，本データにおける発話間の平均時間差が約 1 秒であり，Ishigaki et al. (2021) と比較しても妥当な範囲である．モデルは以降に示す構成要素を主に持ち，生成されたトークン列と正解のトークン列の間のクロスエントロピーを最小化するように学習される．

**Video Encoder** タイミング推定と同様に，タイムスタンプ $(t_s, t_e)$ を持つビデオセグメントを $tau^s_m, \tau^e_m$ に対応付けし，$G$ を用いて $K = \tau^e_m - \tau^s_m + 1$ 個の特徴量のシーケンス $vg_1, \ldots, \boldsymbol{g}_K$ として表現する．

**Text Encoder** 入力テキストを一連のベクトルに対応付けする．文は BERT ベースの単語分割で分割され，特殊なトークンである `CLS` が前置され，最後に `SEP` が追加される．各トークンは学習された位置エンコーディングが組み込まれた $d_m$ 次元の学習された埋め込みにマップされる．トークン化された入力の長さを $L$ とすると, $L+1$ 個のベクトル $vx_0, \ldots, \boldsymbol{x}_L$ の系列ができる．

**Multi-modal Encoder** エンコーダーによって得られたテキストとビデオの両方の特徴を受け取る Transformer．テキストは埋め込まれたテキストトークン列を直接入力し，動画はまず学習可能な線形層を用いて $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K$ を隠れ次元に射影し，さらにこれに学習済みの位置エンコーディング一式を組み合わせて $\tilde{\boldsymbol{g}}_1, \ldots, \tilde{\boldsymbol{g}}_K$ とする．この 2 つのエンコードされたベクトル列を長さ方向に連結し，12 個の attention ヘッドを持つ 12 個のエンコーダブロックに通し，$\boldsymbol{h}_0, \ldots, \boldsymbol{h}_{K+L}$ とする．

**Decoder** cross attention を持つ 6 層 8 ヘッドの Transformer のデコーダー．マルチモーダル文脈に基づく特徴量 $vh_0, \ldots, \boldsymbol{h}_{K+L}$ を受け取り，自己回帰的にキャプションが生成される．

## 5.5 一般動画実況生成におけるタイミング推定実験

5.4 節で設定したサブタスクのうち，タイミング推定の実験について述べる．

### 5.5.1 評価指標

予測された時間間隔が正しいかどうかを判断するために, 先行研究の Krishna et al. (2017) に従って，各正解セグメントとの tIoU を用い，tIoU の値が所定の閾値 (0.3, 0.5, 0.7, および 0.9) 以上であるかかを確認する．すべての閾値の精度と再現率を測定し，それらの値に基づいて計算した F1 スコアの平均を評価指標とする．また，前述の proposal の重複問題における本モデルの影響を調べるために，proposal のセットから可能なすべてのペアを比較し，それらの重複を秒単位で測定し，得られた平均値を評価値とする．

### 5.5.2 実験設定

off-line の動画を C3D (Tran et al., 2015) の特徴量に変換し，$d_v = 500$ で 8 フレームごとに取り出す．動作認識には DBG (Lin et al., 2020) の ActivityNet による事前学習済みモデルを使用し，soft-NMS アルゴリズムによって選択された上位 100 個の proposal を入力として使用した．GRU と LSTM の隠れ状態のサイズは 512 で，$d_p = 100$ とした．モデルは最大 25 エポックで学習され，最適化手法には Adam を使用し，学習率は $4 \times 10^{-4}$ とした．

### 5.5.3 ベースライン

I3D (Carreira and Zisserman, 2017) による特徴と比較する．また, ActivityNet Captions (ANet Cap) の DVC タスクの proposal 抽出に 5.4.1 項で提案した手法を試行し，Mun et al. (2019) の性能と比較する．

表 5.3: タイミング識別モデルの結果.

| データセット | モデル | 動画特徴量 | Backend | Props. | F1 | Overlap (s) |
|---|---|---|---|---|---|---|
| ANet Cap. | Original | C3D | SST | 2.85 | 56.66 | - |
| ANet Cap. | 提案手法 | C3D | DBG | 2.73 | 55.62 | 10.5 |
| 提案データセット | 提案手法 | I3D | DBG | 12.57 | 28.03 | 4.5 |
| 提案データセット | 提案手法 | C3D | DBG | 12.67 | **28.91** | 5.6 |

### 5.5.4 結果

表 5.3 に，Activity Net Captions dataset と本データセットのタイミング識別モデルの結果を示す．ビデオごとの proposal の平均数（Props.）と，識別の F1 スコア（tIoU の閾値 0.3、0.5、0.7、0.9 の平均 F1 スコア），proposal の平均的な重複 (Overlap) を示す．本モデルの平均 F1 スコアは 28.91 であり，C3D の代わりに I3D を用いた場合，若干低下することが分かる．F1 スコアは C3D が I3D を上回るが，proposal の重なりは C3D が上回り，両者の間にトレードオフがあることが分かる．これは，特徴量のサンプリングレートの違いにより，I3D を用いたモデルの方がより細かい時間粒度が得られるためと考えられる．

また、本タスクにおけるモデルの性能は，ANet Cap の平均 F1 スコアは 55.62 と比較して大幅に低い．この性能低下は本タスクが，モデルがより大きなセグメントのセットを選択する必要があるため ANet Cap と比較して難易度が高いためと推測される．一方提案手法は，比較的低いオーバーラップで一連の提案を出力できる．これは，ANet Cap の場合よりも大幅に低いだけでなく，バックエンドモデル (DGB) の平均的な提案のオーバーラップである 7.34 秒よりも低くなる．

このことは，提案したタイミング推定モデルは，TAL を本タスクに適用させる有効な手法であることを示唆している．ただし，この手法は TAL モデルの学習に使用するアノテーションの性質によって精度が左右されうる．

## 5.6 一般ドメイン実況生成における発話生成実験

5.4 節で設定したサブタスクのうち，発話生成実験について述べる．

### 5.6.1 実験設定

I3D による特徴量は，前節の実験で C3D 特徴量よりも良い結果を得たため，本節でも I3D を用いる．動画特徴量は， Rodriguez-Opazo et al. (2021) によって提供されている，25fps，256 × 256 フレームを入力として，平均プーリングの I3D によって $d_v = 1024$ サイズに圧縮した， off-line 動画特徴量を用いる．Transformer ベースのモデルは $d_m = 512$ を使用し，バッチサイズ 8，最適化手法には最大学習率 $10^{-4}$，エポックの 5%の線形アニーリングの Adam を用いて学習する．これらの学習には，NVIDIA V100 GPU を 4 つ使用した．推論時には，ビームサイズ 5 でビームサーチを行う．評価には BLEU-4 (Papineni et al., 2002) を用い，異なるアノテーターのセグメントに対して個別に評価した．

表 5.4: モデル初期化の影響.

| Encoder | Decoder | BLEU-4 |
|---|---|---|
| Random | Random | 2.11 |
| Random | BART-base Dec. | 1.66 |
| BERT-base | Random | 2.07 |
| BERT-base | BART-base Dec. | **2.26** |

### 5.6.2 結果

まず，エンコーダやデコーダを既存の学習済みモデルで初期化する影響を調べる．エンコーダには BERT (Devlin et al., 2019) を，デコーダには BART decoder (Lewis et al., 2020) で初期化する．表 5.4 に示すように，モデルコンポーネントの初期化に事前学習済みのエンコーダーとデコーダーを同時に利用することで，大幅な性能向上が見込める．これらの結果を踏まえ，以降では，すべてのモデルコンポーネントを学習済みモデルで初期化して使用した．

実況が動画のイベントと部分的にしか一致していないというタスクの性質を考慮し，モデルが正しい発話を生成するために，動画およびテキストの前後の状況（コンテキスト）がどのくらい必要かを検討する．両方のモダリティにおいてモデルに与えるコンテキストの量を制御する実験を行った．また，本発話生成モデルがクローズドメイン環境でどのように機能するかを調べるために，Ishigaki et al. (2021) のデータで実験を行った．Ishigaki et al. (2021) の raw 映像を $256 \times 256$ 次元にリサイズし，Carreira and Zisserman (2017) が公開している Kinetics データセットで事前に学習されたモデルによって，I3D 特徴を抽出した．エンコーダの初期化には、日本語 BERT (Suzuki and Takahashi, 2021) 対応のトークナイザーを用いた．また，このデータに対しても，動画・テキストのコンテキストの役割について分析した．

図 5.6 はモデルに与える動画・テキストのコンテキストの量を制御する実験結果をまとめたものである．図中、Y 軸は入力テキスト量，X 軸は入力動画像の時間間隔を表している（順序は厳密ではないが，入力全体の長さと正の相関がある）．例えば、$(t_s, t_e)$ は，正解のアノテーションセグメントから抽出した動画特徴を用いることを意味する．両データセットにおいて，より多くのコンテキストがより良いパフォーマンスをもたらすという状態を明確に観察することができる．また，片方のモダリティでより多くのコンテキストを使うことで，他方のモダリティの使用をある程度補うこともできている．また，発話の開始タイムスタンプより前の特徴のみを受け取る場合，モデルは一般的にあまり良い性能とならなかった．これは、リアルタイム実況生成設定における待機/生成のトレードオフの重要性を示唆している．

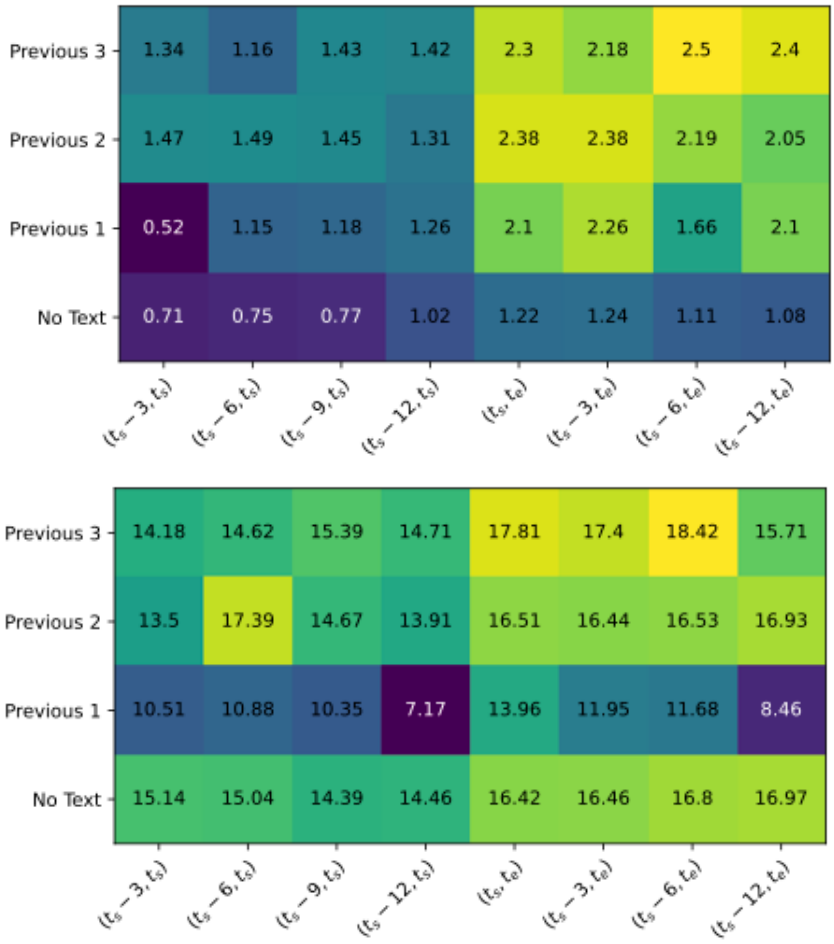



図 5.6: 一般ドメイン設定（上）と Ishigaki et al. (2021) によるクローズドメイン設定（下）のモデルに与える動画・テキスト量を制御した実験結果

最後に，提案する発話生成モデルの性能を，Ishigaki et al. (2021) の性能と比較する．Ishigaki et al. (2021) のアプローチもマルチモーダルモデルであり，入力は 1 秒ごとにキャプチャされた前 10 フレームの動画シーケンス，ゲームプレイから抽出された 10 セットの構造化データ，および前時刻の正解発話とする．入力画像は ViT で符号化し，入出力テキストは文字単位でトークン化し，全体モデルは LSTM を用いた seq2seq で構成されている．表 5.5 から分かるように，本モデルは同程度の動画コンテキストのみを用いた場合（$t_s$ の 9 秒前と 10 秒前では，BLEU-4 で 10.35 と 7.46），大幅に優れた性能を達成できた．これは本提案モデルと動画特徴量抽出手法が妥当であることを示していると考えられる．また，正解の標準区間 $(t_s, t_e)$ の動画特徴を入力した場合，性能は BLEU-4 で 13.96 まで向上するが，$t_s$ 以前の映像を追加してもそれ以上向上しないことも確認した．モデルが追加の動画コンテキストを活用できるのは，テキストコンテキストにもアクセスできる場合のみであり，最

表 5.5: ドメインの性質による精度比較．S,T,V はそれぞれ構造化データ，テキスト，動画コンテキストを表す

| ドメイン | モデル | 利用データ | **BLEU-4** |
|---|---|---|---|
| 一般ドメイン | 提案モデル | V | 1.24 |
| | 提案モデル | V + T | 2.38 |
| クローズドメイン | 提案モデル | V | 13.96 |
| | 提案モデル | V + T | 18.42 |
| | Ishigaki et al. (2021) | V | 7.46 |
| | | S | 23.39 |
| | | S + T | 23.86 |
| | | S + T + V | 24.01 |

終的に BLEU-4 スコアは最大の 18.42 となった．これは，入力動画セグメントのコンテンツと一致しないものを生成する必要があるときに，テキストコンテキストがモデルの決定を助け，また以前に何がすでに言われたかを追跡するためであると考えられる．本モデルと， Ishigaki et al. (2021) で最も良い結果である 24.01 のスコアを得たモデルを比較すると，ドメイン内情報の使用が最も性能向上に寄与しており，実況生成においてドメイン内情報の使用は重要であると言える．

## 5.7 まとめと今後の課題

本章では，一般ドメインの動画に対して実況を生成するタスクを提案した．これは，特定の情報を利用することが可能なクローズドメインにのみ焦点を当ててきたこれまでの研究を大幅に拡張するものである．さらに，ドメインを限らないオープンドメインの課題とすることで，ある事象に対して言葉の広がりを許容する課題となった．本課題は，新たに導入したタスクのため，様々なドメインにおける人間の様々な行動を含む 6K 以上の動画と実況テキストのデータセットを作成した．本章で提案した現在の手法の長所と限界を示す研究とともに，本課題へのモデルを提案した．この結果を踏まえ，今後は Ishigaki et al. (2021) による構造化データのようなドメイン固有の情報も取り込むことができるクローズドメイン・マルチモーダル Transformer モデルを開発できると考えている．また，一般ドメインの実況生成には，最近 DVC に対して提案された end-to-end モデル (Wang et al., 2021) の適用や，on-line 設定による課題解決に取り組むべきと考えている．

本章の実況は，動画の内容を正しく途切れることなく説明しているかを重視して収集している．将来的には，人を惹きつけるような娯楽に寄った実況や，逆に動画内容の専門性

をより反映した実況なども考えられる.

# 第6章　実データを用いたdata-to-text

本章では，非言語から言語を生成する課題として，実データを用いた data-to-text を扱う．特に，日経平均株価データの概況テキスト生成に着目し，実データを用いた非言語から言語を生成する課題において発生する問題について 2 つ取り上げる．それらの問題を解決する手法について提案・実験し，評価する．

## 6.1　概要

金融や医療，情報通信などの多くの分野において，様々な形式の非言語データを取り扱う機会が増えてきている．大規模で複雑なデータを専門知識のない人が解釈できるように，データを説明するテキストを自動的に生成する技術の必要性が高まっており，近年は様々なデータを題材に，Encoder-Decoder モデルを用いて end-to-end の学習を行うことで，高い生成性能を発揮している (Puzikov and Gurevych, 2018; Liu et al., 2018a; Iso et al., 2019).

data-to-text のデータセットは，実世界で獲得されたデータを用いたデータセットと，入出力をコントロールして人手で作成されたデータセットが存在する．人手で作成されたデータセットの場合，データとテキストが過不足なく組み合わさっている. 一方、実世界で得られたデータとテキストを使ってデータセットを作成する場合，データとテキストのアライメントは，人手によるアノテーションや，データとテキストに付随する他のデータによって取得される．アライメントが不十分な場合，データとテキストの参照箇所に不整合が生じ，それにより誤りが発生することがある (Taniguchi et al., 2019)．また，人手で作成されたデータセットの場合には入力データから出力テキストを推定可能であるように調節してデータセットを作ることができるが，実世界で得られたデータとテキストを使ってデータセットを作成する場合，入出力をコントロールできないため、入力データから出力テキストを推定できない場合がある．その場合，入力から出力を推定できるように、例えばラベルなどを用いて追加でデータを入力する必要がある．人手で作成されたデータセットと実データを用いたデータセットはその課題の難易度も異なる．例えば，人手で作成されたデータセットを用いた E2E NLG Challenge において，2023 年現在最も高い BLEU スコアは Kale and Rastogi (2020) による 68.60 だが，実データを用いて作成された WikiBio の生成課題では，同様の課題でありながら，最も高い BLEU スコアは Liu et al. (2018b) による 44.89 である．このことからも，難易度の差は明確であろう．本章では日経平均株価データ

からの概況テキスト生成を対象に，実データを利用する際に生じる問題について分析し，解決策を探る．

## 6.2 関連研究

入力データを説明するテキストを生成するタスクは data-to-text と呼ばれ，さまざまの領域で研究されている．例えば，時系列の気象情報から天気予報テキストを自動生成する研究 (Belz, 2007; Angeli et al., 2010) や，臨床データから概況テキストを生成する研究 (Belz, 2007; Angeli et al., 2010)，スポーツの試合データやスコア情報から試合内容をテキスト生成する研究 (Liang et al., 2009) などが挙げられる．伝統的に，data-to-text は *"what to say*（何を言うか）*"* の *content selection* と，*"how to say*（どのように言うか）*"* の *surface realization* の 2 つのサブタスクに分けられる (Kukich, 1983; Goldberg et al., 1994)．さらに，Reiter and Dale (1997) は，*micro planning* をこの 2 つのサブタスクに加えて，3 つのサブタスクとした．Data-to-text の初期段階では，主に *surface realization* のタスクに取り組み，テンプレートを用いる手法 (van Deemter et al., 2005) や，人手で作成された特徴量によって統計的に学習されたモデルを用いる手法 (Belz, 2008; Konstas and Lapata, 2012) が提案された．

一方，近年では金融，製薬，通信など様々な業界で，様々な種類の大規模データを扱う機会が増えており，データとテキストの大規模なペアデータを基に対応関係を自動的に学習し，データの内容をテキストで生成する手法への関心が高まっている．特に，上記のサブタスクを一度に解決できるニューラルネットワークを用いた手法に注目が集まっており，その中でも Encoder-Decoder モデルの利用が有用であるとされている (Mei et al., 2016; Lebret et al., 2016)．

Murakami et al. (2017) は Data-to-text の一タスクとして，日経平均株価の市況コメントを生成するタスクを取り上げ，時系列数値データから多様な特徴を抽出し，データの概要をテキスト化する手法を提案した．市況コメントなどの時系列数値データの概況テキストでは，「上がる」，「下がる」といった単純な特徴だけが表出されるわけではない．過去のデータの履歴や，テキストが書かれる時間帯によって言及すべき内容は様々である．また，数値の時系列データの場合，時系列中の数値や，過去との差分を計算した値が言及されることが多い．株価の市況コメントにおけるこれらの特性を踏まえ，データから多様な特徴を自動抽出し，テキスト化するためのエンコード／デコード手法を提案した．まず，「続落」，「上げに転じる」といった時系列株価データの過去の履歴や変化を捉えるために，株価の短期的および長期的な時系列データを使用した．次に，「前引け」，「大引け」といった市況コメントが記述される時間帯に依存する表現を生成するために，デコード時に時刻情報を導入した．

ニューラルネットワークによるテキスト生成は，流暢にテキストを記述できる反面，正確

なエンティティーや数値を記述することはできない．そこで，入力から直接単語をコピーできるコピー機構 (Vinyals et al., 2015a; Gu et al., 2016) の利用により，表を基にした条件付き言語生成 (Yang et al., 2017) や Wikipedia のインフォボックスからのバイオグラフィー生成 (Lebret et al., 2016; Chisholm et al., 2017; Sha et al., 2018; Liu et al., 2018a)，スポーツのスコアボードからの試合サマリーの生成 (Wiseman et al., 2017; Li and Wan, 2018; Puduppully et al., 2019) などでは成果を上げている．しかし，コピー機構では入力に現れる表層的な内容しか生成できず，算術演算を必要とするコンテンツは生成できない．一方，Joulin and Mikolov (2015) や Neelakantan et al. (2016) は，現在のニューラルモデルでは，加算や比較などの算術演算をニューラルネットワークを用いた手法で学習することが困難であると示している．市況コメントなどの数値の時系列データの内容を言及する場合，時系列中の数値や，過去との差分を計算した値が言及されることが多い．そこで，Murakami et al. (2017) は「19,386 円」，「100 円」といった株価の終値や前日からの変動幅などの数値を市況コメントで言及するために，価格を算術演算に置き換えることで，時系列中の数値や，過去との差分を計算した値の生成を行えるようにした．数値の出力は推定した演算操作と入力の時系列株価データを用いて計算することで行う．次節では，Murakami et al. (2017) について，詳しく説明する．

## 6.3 日経平均株価の概況テキスト生成

Murakami et al. (2017) は，Encoder-Decoder モデル (Sutskever et al., 2014) に基づいて，日経平均株価データとその動きに言及しているニュースヘッドラインを例に，時系列数値データから概況テキストを生成するモデルを開発した．

### 6.3.1 時系列株価データから市況コメントを自動生成するモデル

Murakami et al. (2017) は，概況テキストが発表された時刻を基準に，長期的な変動を捉えるための直近 $M$ 取引日分の終値データ $\boldsymbol{x}_{\text{long}} = (x_{\text{long},1}, x_{\text{long},2}, \ldots, x_{\text{long},M})$ と，短期的な変動を捉えるために 5 分足で収集した株価データの直近の $N$ 個の市況データを $\boldsymbol{x}_{\text{short}} = (x_{\text{short},1}, x_{\text{short},2}, \ldots, x_{\text{short},N})$ を取得し，入力として用いる．2 種類のデータに対して，それぞれ以下の前処理を適用する：

$$x_{\text{std}_\text{i}} = \frac{x_\text{i} - \mu}{\sigma}, \tag{6.1}$$

$$x_{\text{move}_\text{i}} = x_\text{i} - r_\text{i}, \tag{6.2}$$

$$x_{\text{norm}_\text{i}} = \frac{2 \times x_{\text{move}_\text{i}} - (\bar{x}_{\max} + \bar{x}_{\min})}{\bar{x}_{\max} - \bar{x}_{\min}}. \tag{6.3}$$

1 つ目は式 (6.1) で表される平均値と標準偏差を用いて標準化する手法であり，2 つ目は，前日の終値からの価格の変動を捉えるために，式 (6.2) によってそれぞれの入力データに対し

て前取引日の終値 $r_{\mathrm{i}}$ からの差分を計算し，差分の最大値 $\bar{x}_{\max}$，最小値 $\bar{x}_{\min}$ を用いて，式 (6.3) によって $[-1, 1]$ へ正規化を行う手法である．これらの前処理を，$\boldsymbol{x}_{\mathrm{long}}$ および $\boldsymbol{x}_{\mathrm{short}}$ に適用し，得られたベクトルをそれぞれ $\boldsymbol{x}^{\mathrm{std}}_{\mathrm{long}}$，$\boldsymbol{x}^{\mathrm{move}}_{\mathrm{long}}$，$\boldsymbol{x}^{\mathrm{std}}_{\mathrm{short}}$ および $\boldsymbol{x}^{\mathrm{move}}_{\mathrm{short}}$ とする．

エンコーダーでは，multi-layer perceptrons (MLP)[1]を用いて，それぞれ以下を得る：

$$\boldsymbol{h}_{\mathrm{long}} = \left[\mathrm{MLP}(\boldsymbol{x}^{\mathrm{std}}_{\mathrm{long}}); \mathrm{MLP}(\boldsymbol{x}^{\mathrm{move}}_{\mathrm{long}})\right], \tag{6.4}$$

$$\boldsymbol{h}_{\mathrm{short}} = \left[\mathrm{MLP}(\boldsymbol{x}^{\mathrm{std}}_{\mathrm{short}}); \mathrm{MLP}(\boldsymbol{x}^{\mathrm{move}}_{\mathrm{short}})\right]. \tag{6.5}$$

これらを結合し，市況データをエンコードした隠れ状態 $\boldsymbol{m}$ を得る：

$$\boldsymbol{m} = \boldsymbol{W}_m \left[\boldsymbol{h}_{\mathrm{long}}; \boldsymbol{h}_{\mathrm{short}}\right] + \boldsymbol{b}_m. \tag{6.6}$$

デコーダは LSTM を用い，デコーダの初期値 $\boldsymbol{s}_0$ として，前述した隠れ状態 $\boldsymbol{m}$ を用いる．時刻 $t$ におけるデコーダの隠れ層の状態 $\boldsymbol{s}_t$ は，ニュースヘッドラインの配信時刻 (9 時や 13 時などの 1 時間刻みの数値) を埋め込んだ時間帯情報埋め込みベクトル $\boldsymbol{t}$ と，直前の単語埋め込み $\boldsymbol{w}_{t-1}$，直前の隠れ層の状態 $\boldsymbol{s}_{t-1}$ を用いて次のように計算される:

$$\boldsymbol{s}_i = \mathrm{LSTM}([\boldsymbol{t}; \boldsymbol{w}_{i-1}], \boldsymbol{s}_{i-1}). \tag{6.7}$$

### 6.3.2 数値の汎化演算タグ

概況テキストでは，市場価格そのものや、差分や丸めなどの算術演算を行った数値が記述されることが多い．このような演算を伴う数値をデコード時に生成できるようにするために， Murakami et al. (2017) では演算タグ (表 6.1) を導入し，値を得るためにどの演算を行うべきかを指定している．デコーダが汎化タグを生成すると、モデルはそのタグを対応する算術演算によって得られた値に置き換え，最終的に数値を含むテキストを生成する．Murakami et al. (2017) では，最後のタイムステップの価格 $x_{\mathrm{short},1}$ を $z$，前日の終値 $x_{\mathrm{long},1}$ と $z$ の差を $\Delta$ とする．以下の文章を例に，学習データのテキスト演算タグを用いたの前処理方法を説明する．

($a$) 日経平均、続伸で始まる。9 円高の 17024 円

前処理では，まず表 6.1 中の全ての算術演算を行い，各演算タグに対応する数値を計算する．例えば，テキスト ($a$) が配信された時刻を基準として，前日の終値 $x_{\mathrm{long},1}$ が 17,015，最新の価格 $x_{\mathrm{short},1}$ が 17,024 の場合，演算タグ<operation1>に対応する数値は「9」となる．次に，各演算タグに対応する数値とテキスト (a) 中の数値「9」,「17,015」を比較し，テキ

[1] Murakami et al. (2017) では，MLP, RNN,convolutional neural networks (CNN) および long short-term memory networks (LSTM) を比較実験した．MLP を用いた場合に最も精度が良かったため，本論文では MLP を用いる．

表 6.1: 演算タグと対応する算術演算.

| 演算タグ | 算術演算 |
|---|---|
| <operation1> | $\Delta$ を返す |
| <operation2> | $\Delta$ を 10 の位で切り下げ |
| <operation3> | $\Delta$ を 100 の位で切り下げ |
| <operation4> | $\Delta$ を 10 の位で切り上げ |
| <operation5> | $\Delta$ を 100 の位で切り上げ |
| <operation6> | $z$ を返す |
| <operation7> | $z$ を 100 の位で切り下げ |
| <operation8> | $z$ を 1,000 の位で切り下げ |
| <operation9> | $z$ を 10,000 の位で切り下げ |
| <operation10> | $z$ を 100 の位で切り上げ |
| <operation11> | $z$ を 1,000 の位で切り上げ |
| <operation12> | $z$ を 10,000 の位で切り上げ |

スト $(a)$ の数値それぞれに最も近い数値を計算した演算タグを求める．最後に導出した演算タグで正解の数値を置換して，以下の前処理済みテキスト $(b)$ を獲得する．

$(b)$ 日経平均、続伸で始まる。<operation1>円高の<operation6>円

数値を含むテキストを生成する場合，まずデコーダーによってテキスト $(c)$ のような演算タグを含むテキストが生成される．

$(c)$ 日経平均、反落で始まる。下げ幅<operation2>円超、<operation7>円台

この時前日の終値 $x_{\mathrm{long},1}$ が 14,612，最新の価格 $x_{\mathrm{short},1}$ が 14,508 の場合，<operation2> は演算により「100」, <operation7> は「14,500」と計算される．これらの数値でテキスト $(c)$ の演算タグを置換して，最終的な数値を含むテキスト $(d)$ が出力される．

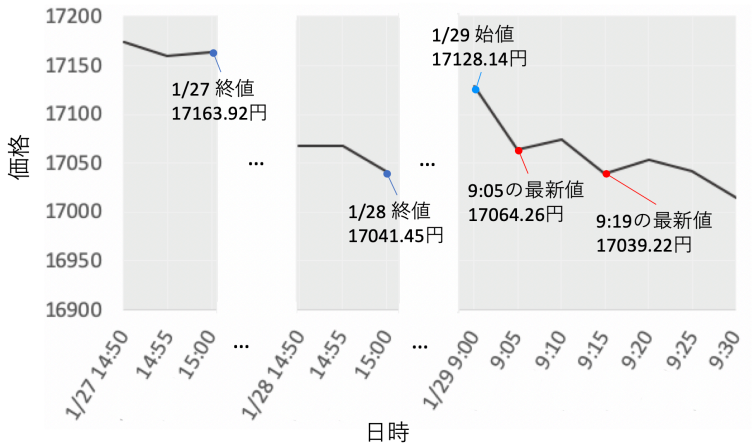
$(d)$ 日経平均、反落で始まる。下げ幅 100 円超、14, 500 円台

## 6.4 参照時間の不整合

Murakami et al. (2017) は, 概況テキストが発表された時刻を基準に, 長期的な変動を捉えるための直近 $M$ 取引日分の終値データ $\boldsymbol{x}_{\mathrm{long}} = (x_{\mathrm{long},\,1}, x_{\mathrm{long},\,2}, \ldots, x_{\mathrm{long},\,M})$ と, 短期的な変動を捉えるために 5 分足で収集した直近の $N$ 個の市況データ $\boldsymbol{x}_{\mathrm{short}} = (x_{\mathrm{short},\,1}, x_{\mathrm{short},\,2}, \ldots, x_{\mathrm{short},\,N})$ を取得し，入力とした．そのため，記事が発表された時刻（以降，配信時刻）と記事を作成

した時刻（以降，参照時刻）の不整合は無視されてきた．例えば，図 6.1 に示した（I）から（Ⅲ）の文章は，全て 1 月 29 日 9 時を参照時刻とする概況テキストであり，概況テキストでの動向内容と参照時刻での値動きは一致している．（I）は，配信時刻と参照時刻が同じため，概況テキストの動向内容と記事の配信時刻を基準とした値動きは一致している．一方，（Ⅱ）および（Ⅲ）では，配信時刻と参照時刻が異なるため，概況テキストの動向内容と配信時刻での値動きは異なっている．

そこで，テキストが発表された時刻の直近のデータだけでなく，それ以前の時間帯で取得できるデータを用いて，参照時刻の不整合解消を目指す．また，データの傾向と時間帯の情報から，参照時刻を取得する機構を組み込むことで，精度向上に取り組む．




| | 配信時刻 | 配信時刻の値動き | 概況テキスト（動向内容） |
|---|---|---|---|
| （I） | 09:00 | 反発，86 円高 | 日経平均、反発で始まる　86 円高 |
| （Ⅱ） | 09:05 | 反発，22 円高 | 日経平均、反発で始まる　86 円高、原油高を好感　トヨタ高い |
| （Ⅲ） | 09:19 | 続落，2 円安 | 東証寄り付き、反発　原油高を好感、ファナックは大幅安 |

図 6.1: 日経平均株価の値動きと，1 月 29 日 9 時を参照時刻とする概況テキスト．

### 6.4.1 Multi-timestep 構造と Copy 機構を用いたテキスト生成

Murakami et al. (2017) は，タスクを単純化するため参照時刻と配信時間が同一であるという仮定を置き，基本的なエンコーダ・デコーダの構造を持つモデルを提案している．しかし、実際の参照時刻と配信時刻の間には時間差があるため、この仮定は現実的ではない．そこで、このようなアライメントによる問題を解決するために，Multi-timestep 構造を導入し，モデルを拡張する．図 6.2 にその概要を示す．先行研究の時間差による情報不足を補うために，概況テキストが発表された時刻から取得できる直近のデータだけでなく，それ

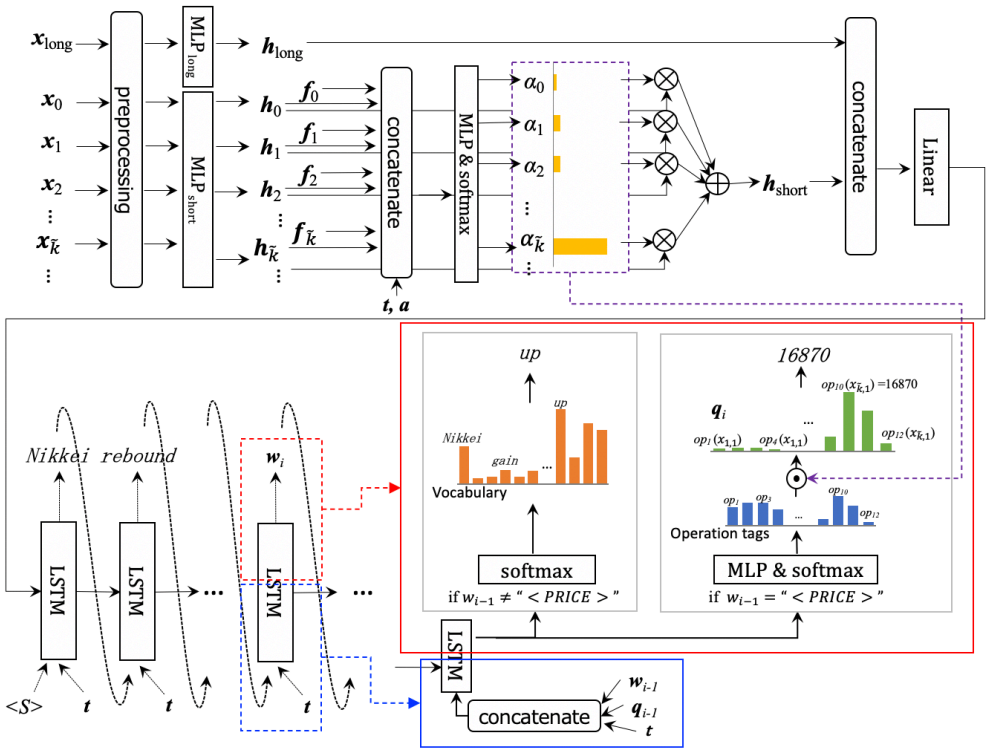



図 6.2: 提案モデルの全体概要図.

以前の時刻で取得できるデータを入力に追加する．各追加ベクトルは、配信時刻の代わりに $k$ 前のタイムステップから始まる $\boldsymbol{x}_{\text{short}}$ に相当する．これにより、実際のイベント時刻を潜在変数として扱うことができる．その上で、データの傾向と時間帯の情報から，参照時刻を取得するできるように，デコーダにアテンション付き Copy 機構を導入し，ノイズのある学習データからデータとテキストの対応関係を容易に学習できるようにする．

### Multi-timestep Encoder

入力を長期的な変動を捉えるための直近 $M$ 取引日分の終値データ $\boldsymbol{x}_{\text{long}}$ と，短期的な変動を捉えるための 5 分足で収集した市況データの直近 $N$ 個の数値 $\boldsymbol{x}_0$ に加え，1 ステップ前から $n$ ステップ前に取得できる各 $N$ 個の数値 $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ とする．Murakami et al. (2017) と同様に，各入力に前処理をおこない，$\boldsymbol{x}^{\text{std}}_{\text{long}}$，$\boldsymbol{x}^{\text{move}}_{\text{long}}$，$\boldsymbol{x}^{\text{std}}_0$，$\boldsymbol{x}^{\text{std}}_1,\ldots,\boldsymbol{x}^{\text{std}}_n$ および $\boldsymbol{x}^{\text{move}}_0$，$\boldsymbol{x}^{\text{move}}_1,\ldots,\boldsymbol{x}^{\text{move}}_n$ を得る．これらのベクトルにそれぞれ MLP を用いて，$\boldsymbol{h}_{\text{long}}$ と，$\boldsymbol{h}_0, \boldsymbol{h}_1,\ldots,\boldsymbol{h}_n$ を得る．

$\boldsymbol{x}_k$ の重要度を得るために，新たな埋め込みベクトル $\boldsymbol{a}$，$\boldsymbol{f}_k$ と，時間帯情報埋め込みベク

トル $\boldsymbol{t}$ を導入する．$\boldsymbol{a}$ は概況テキストの種類に関するタグ[2]，$\boldsymbol{f}_k$ は $\boldsymbol{x}_k$ の最新値が取得された 5 分間隔の時刻を表す．重要な $\boldsymbol{x}_{\tilde{k}}$ は，市況データの価格変動（$\boldsymbol{h}_{\tilde{k}}$）もしくは，配信時間（$\boldsymbol{f}_{\tilde{k}}$）のどちらかで主に決定されるが，概況テキストの種類 $\boldsymbol{a}$ と時間帯 $\boldsymbol{t}$ によって，そのどちらを用いるかは異なる．例えば，配信時刻が市場の開始時刻と同じ 9 時で，決まった時刻に配信されるコメントの場合，コメントは「日経平均、15,430 円で始まる。」のように，通常 9 時の価格に言及する．この場合，重要な $\boldsymbol{x}_{\tilde{k}}$ は主に $\boldsymbol{f}_{\tilde{k}}$ でエンコードされた時間によって決定される．つまり，決まった時刻に配信されるコメントの場合，配信時間の変動に関わらず，参照時刻はむしろ固定されている．一方，配信時刻が市場の開始時刻と同じであっても，決まった時刻に配信されるコメントではない場合，単なる 9 時時点の価格ではなく，例えば「前日の終値より 100 円以上高くなった。」のように，何か特徴的な値動きについて言及する．この場合，重要な $\boldsymbol{x}_{\tilde{k}}$ は価格変動（$\boldsymbol{h}_{\tilde{k}}$）によって決定される．
まず，$\boldsymbol{h}_k (k = 0, 1, \ldots, n)$ をそれぞれの $\boldsymbol{x}_k$ の最新値が取得された時刻 $\boldsymbol{f}_k$ と結合し，さらに，時間帯情報埋め込みベクトル $\boldsymbol{t}$ と，概況テキストの種類に関するタグ $\boldsymbol{a}$ を結合して MLP を行う．その結果で softmax することで $\boldsymbol{x}_k$ の重要度を表す $\alpha_k$ を得る．

$$\boldsymbol{e} = \mathrm{MLP}\left([\boldsymbol{t}; \boldsymbol{a}; \boldsymbol{f}_0; \boldsymbol{h}_0; \ldots; \boldsymbol{f}_n; \boldsymbol{h}_n]\right). \tag{6.8}$$

$$\alpha_k = \exp(e_k) / \sum_{j=0}^{n} \exp(e_j). \tag{6.9}$$

得られた $\alpha_k$ と用いて，$\boldsymbol{h}_{\mathrm{short}}$ は

$$\boldsymbol{h}_{\mathrm{short}} = \sum_{k=0}^{n} \alpha_k \boldsymbol{h}_k. \tag{6.10}$$

のように，$\boldsymbol{h}_k (k = 0, 1, \ldots, n)$ の加重和とし，式 6.6 と同様に，エンコードした隠れ状態 $\boldsymbol{m}$ を $\boldsymbol{h}_{\mathrm{long}}$ と $\boldsymbol{h}_{\mathrm{short}}$ から得る．

$$\boldsymbol{m} = \boldsymbol{W}_m \left[\boldsymbol{h}_{\mathrm{long}}; \boldsymbol{h}_{\mathrm{short}}\right] + \boldsymbol{b}_m. \tag{6.11}$$

これをデコーダの初期値 $\boldsymbol{s}_0$ として用いる．

### Copy 機構を用いた Decoder

Murakami et al. (2017) は，演算タグに最後のタイムステップの価格 $x_{\mathrm{short},1}$ と前日の終値 $x_{\mathrm{long},1}$ のみを単純に用いた．この単純化では、配信時刻とイベント時刻の不一致の可能性が無視され，その結果，参照時刻での数値とは異なる数値が生成される．そこで、参照時刻を予測して表 6.1 の操作でデコード時に各数値を適切に生成できるように，Copy 機構 (Gu

[2]決まった時刻に配信されるコメントと，それ以外の 2 種類

et al., 2016) を導入し，モデルを拡張する．具体的には，各 $\boldsymbol{x}_k$ の最初の値（最新の価格）であるデータ点 $x_{\tilde{k},1}$ を用いて計算され，数値演算は各 $(o,k)$ に対して $o \in \{op_1, \cdots, op_{12}\}$ で表す．数値の生成は，$(o,k)$ の選択と演算の実行にとどめる．

$(o,\tilde{k})$ によって決まる，数値の確率分布を求める．このとき、$(o,k)$ の組が複数存在し，同じ値になる可能性がある．例えば，ある入力 $x_{\tilde{k},1} = z = 3200$ の場合，$op_6$ と $op_7$，$op_{10}$ は全て同じ値（3200）となる．

実際には，生成する数値 $w_i$ は Attention 機構によって得られたエンコーダの重み $\alpha_k$ に従って $(o,k)$ のスコアの重み付き和 $o(x_{\tilde{k},1}) = w_i$ である．

一般的な Copy 機構とは異なり，デコーダー全体で固定の Attention 重み $\alpha_k$ を用いる．これは，生成目的のヘッドラインテキストは言及する事柄が 1 つのテキストを通して変化しないためである．つまり，重要な $\boldsymbol{x}_k$ は 1 つの概況テキスト中で変化することはない．我々のモデルは、この Copy 機構によって、テキスト中のすべての数値を生成する。そのために、モデルのボキャブラリーから数値を除外する。コピーモードと非コピーモードを切り替えるために、学習データのすべての数値の前に挿入され、次のトークンが価格値であることを示す特別なトークン *“<PRICE>”* を追加する．これにより，*“<PRICE>”* の次の単語が数値であることを示す．*“<PRICE>”* を利用して，時刻 $i$ に対象単語 $w_i$ が生成される各条件付き確率を次のように定義する：

$$p(w_i|w_{<i}, \boldsymbol{m}) = \begin{cases} p_{\text{COPY}}(w_i|w_{<i}, \boldsymbol{m}), & (w_{i-1} = \text{“<PRICE>”}) \\ p_{\text{GEN}}(w_i|w_{<i}, \boldsymbol{m}), & (\text{otherwise}) \end{cases} \tag{6.12}$$

この時，$p_{\text{GEN}}(w_i|\cdot)$ は *generation mode* から，$p_{\text{COPY}}(w_i|\cdot)$ は *copy mode* からそれぞれ得られる．この手法は， See et al. (2017) によって提案された *Pointer-generator network* を参考にした．$p_{\text{GEN}}(w_i|\cdot)$ と $p_{\text{COPY}}(w_i|\cdot)$ は次のように定義する：

$$p_{\text{GEN}}(w_i|\cdot) = [\text{softmax}(\boldsymbol{W}_v \boldsymbol{v}_i + \boldsymbol{b}_v)]_{w_i} \tag{6.13}$$

$$p_{\text{COPY}}(w_i|\cdot) = \sum_{\tilde{k}, o: o(x_{\tilde{k},1}) = w_i} q(\tilde{k}, o) \tag{6.14}$$

$$q(\tilde{k}, o) = \alpha_k \cdot [\text{softmax}(\boldsymbol{W}_c \boldsymbol{c}_i + \boldsymbol{b}_c)]_o. \tag{6.15}$$

$\boldsymbol{v}_i$ と $\boldsymbol{c}_i$ はどちらも LSTM の $i$ 目の出力 $\boldsymbol{s}_i$ で次のように定義される:

$$\boldsymbol{s}_i = \text{LSTM}([\boldsymbol{t}; \boldsymbol{w}_{i-1}; \boldsymbol{q}_{i-1}], \boldsymbol{s}_{i-1}) \tag{6.16}$$

$$\boldsymbol{v}_i = \boldsymbol{W}_h \boldsymbol{s}_i + \boldsymbol{b}_h \tag{6.17}$$

$$\boldsymbol{c}_i = \text{MLP}(\boldsymbol{s}_i). \tag{6.18}$$

$\boldsymbol{t}$ は 6.3.1 節で定義した通り，時間帯情報埋め込みベクトルである．式 (6.7) の LSTM に $\boldsymbol{q}_i$ を追加しており，各 $\boldsymbol{q}_i$ は $q(\tilde{k}, o)$ である．このベクトルを追加することで，算術演算結果を $\boldsymbol{s}_i$ に直接保持させずに適切に伝播させることができる．

### 6.4.2 実験設定

#### データセット

実験には，時系列データとして，Thomson Reuters DataScope Select[3]から，2010 年 12 月から 2016 年 9 月までの期間の日経平均株価指数の時系列データを収集し，実験に使用した．概況テキストとして，日経 QUICK ニュース社が提供している日経平均株価に言及しているニュース記事のヘッドラインを使用した．2010 年 12 月から 2015 年 9 月の期間のデータを学習データ（15,035 件），2015 年 10 月から 2016 年 3 月の期間のデータを開発データ（1,759 件），2016 年 4 月から 2016 年 9 月の期間のデータを評価データ（1,695 件）として使用した．実験に使用したデータについて，概況テキストに日経平均株価の数値の動向を説明する表現[4]が含まれているか，含まれている場合には，その表現と取得したデータの数値の動向が一致しているかを調査した．その統計値を表 6.2 に示す．

表 6.2: 使用したデータの統計値

| データ種類 | Movement 動向表現 | | | 合計 |
|---|---|---|---|---|
| | なし | あり | | |
| | | 一致 | 不一致 | |
| 学習データ | 3,522 | 11,172 | 341 | 15,035 |
| 開発データ | 378 | 1,346 | 35 | 1,759 |

#### 実験設定

5 分足で収集した記事配信時刻直近の 62 個の市況データ $\boldsymbol{X}_{\text{short-latest}}$，直近 7 取引日分の終値データ $\boldsymbol{X}_{\text{long}}$，および概況テキストのペアを使用する．また記事配信時刻以前のデータとして，1 ステップ前 $\boldsymbol{X}_{\text{short-1step}}$ から 6 ステップ前 $\boldsymbol{X}_{\text{short-6step}}$ までを段階的に用いる．なお，時系列株価データからテキスト中で言及された価格を適切に出力するために，Murakami et al. (2017) と同様に，概況テキスト中の数値表現を演算タグに置換した．株価等の時系列数値データをベクトルへ変換する各 MLP の隠れ状態の次元は 256，これらの隠れ状態を結合した後，線形変換により次元を 256 とした．デコーダは 1 層の LSTM で，隠れ状態の次元は 256，単語埋め込みベクトルの次元は 128，時間帯情報埋め込みベクトルの次元は 64，時刻情報埋め込みベクトル $\boldsymbol{f}_k$ の次元は 80, 記事タグ埋め込みベクトル $\boldsymbol{a}$ の次元は 64 とした．各パラメータの最適化手法には Adam (Kingma and Ba, 2015) を使用し，学習率は $1 \times 10^{-4}$ とした．ミニバッチサイズは 100，エポック数は 150 とし，学習時に開発デー

[3] https://hosted.datascope.reuters.com/DataScope/
[4] 上方向の動向：「続伸，反発，上げ」，下方向の動向：「続落，反落，下げ」

タで計算された BLEU が連続して下がった場合には学習を終了し，各エポックのモデルの中で，開発データに対する BLEU が最大となったモデルを使用して，評価を行う．

### 6.4.3 評価方法

評価指標には，実際の株価の概況テキストと生成されたテキストの一致度合いを測る目的として BLEU (Papineni et al., 2002) を使用する．また，正解概況テキストおよび生成した概況テキストの，日経平均株価の短期的・長期的な動向を説明する表現と，配信時刻直近の数値データの動向の一致・不一致を取得して評価する方法を新たに提案する．

開発データの $i$ 番目の事例とモデルの生成文を合わせて $(\boldsymbol{x}_i, \boldsymbol{w}_i^{gold}, \boldsymbol{w}_i^{pred})$ とし，次のように定義する:

$$move_{text}(\boldsymbol{w}) = \begin{cases} \text{上方向} & (\boldsymbol{w} \in \{\text{続伸，反発，上げ}\}) \\ \text{下方向} & (\boldsymbol{w} \in \{\text{続落，反落，下げ}\}) \\ \text{なし} & (otherwise) \end{cases}$$

$$move(\boldsymbol{x}) = \begin{cases} \text{上方向} & (\boldsymbol{x} \text{ の直近の数値動向} > 0) \\ \text{下方向} & (\boldsymbol{x} \text{ の直近の数値動向} < 0) \\ \text{なし} & (otherwise) \end{cases}$$

$$\begin{aligned} \boldsymbol{C_{gold}} &= \{i | move_{text}(\boldsymbol{w}_i^{gold}) = move(\boldsymbol{x}_i)\} \\ \boldsymbol{C_{pred}} &= \{i | move_{text}(\boldsymbol{w}_i^{pred}) = move(\boldsymbol{x}_i)\} \\ \boldsymbol{D_{gold}} &= \{i | move_{text}(\boldsymbol{w}_i^{gold}) \neq move(\boldsymbol{x}_i)\} \\ \boldsymbol{D_{pred}} &= \{i | move_{text}(\boldsymbol{w}_i^{pred}) \neq move(\boldsymbol{x}_i)\} \end{aligned}$$

これらを用いて，次の式により評価する:

$$\begin{aligned} \text{一致}_{precision} &= \frac{|\boldsymbol{C_{gold}} \cap \boldsymbol{C_{pred}}|}{|\boldsymbol{C_{pred}}|} \\ \text{一致}_{recall} &= \frac{|\boldsymbol{C_{gold}} \cap \boldsymbol{C_{pred}}|}{|\boldsymbol{C_{gold}}|} \\ \text{不一致}_{precision} &= \frac{|\boldsymbol{D_{gold}} \cap \boldsymbol{D_{pred}}|}{|\boldsymbol{D_{pred}}|} \\ \text{不一致}_{recall} &= \frac{|\boldsymbol{D_{gold}} \cap \boldsymbol{D_{pred}}|}{|\boldsymbol{D_{gold}}|}. \end{aligned}$$

この評価指標は，モデルが配信時刻直近の数値データの動向だけでなく，それ以前のステップにアノテーションする能力の評価と見なすことができる．データセット中の概況テキストの動向表現と配信時刻直近の数値データの動向の一致・不一致の統計値を表 6.2 に示す．

BLEU による評価は開発データおよび評価データに対して行い，概況テキストの動向表現と数値データとの比較による評価は開発データに対して行った．なお，学習時の seed を 0，5，10，50，100，500 の 6 種類とし，各モデルによる評価結果の平均を 6.4.4 節の実験結果として用いた．

### 6.4.4 実験結果

表 6.3: BLEU (%)

| | 開発 | 評価 | 開発 | 評価 | 開発 | 評価 |
|---|---|---|---|---|---|---|
| ベースライン | 21.37 | 21.30 | - | - | - | - |
| **Multi-timesteps** | | | **+Attention** | | **+Copy** | |
| $n = 0$ | 21.63 | 22.66 | 21.33 | 22.38 | 28.16 | 28.68 |
| 1 | 21.59 | 22.74 | 21.43 | 22.25 | 27.90 | 28.54 |
| 2 | 21.64 | 23.03 | 21.20 | 22.47 | **28.31** | **28.98** |
| 3 | 21.82 | 23.11 | 21.02 | 21.94 | 27.29 | 27.75 |
| 4 | 21.68 | 22.92 | 20.59 | 21.71 | 28.13 | 28.93 |
| 5 | 21.73 | 22.89 | 20.71 | 21.74 | 27.29 | 27.78 |
| 6 | 21.73 | 22.66 | 20.51 | 21.62 | 26.68 | 27.25 |

BLEU での評価結果を表 6.3 に，動向一致・不一致による評価結果を表 6.4 に示す．表 6.3 によると，提案手法のうち，Copy 機構を導入した場合にほとんどの場合ベースラインを上回った．評価データセットでは，全ての場合でベースラインを上回った．特に，Copy 機構を用いた $n = 2$ の場合に最も高いスコアとなり，ベースラインと比べて 7.68 ポイント高くなった（表 6.3 の **bold** テキストを参照のこと）．一方で，デコーダはベースラインと同じにし，エンコーダに注意を加える（+ Attention）だけでは，効果がないことも示している．以上により，Attention 機構を適用しただけでは，データとテキストの対応関係を正しく得ることはできないが，Copy 機構を適用すれば，対応関係を正しく得ることができることがわかる．なお $n = 0$ から 6 の結果を比べると，大きくポイントが変わらないことから，ステップ数を増やしても必ずしも BLEU スコアの向上には寄与しないと言える．

さらに表 6.4 によると，提案手法のほとんどがベースラインを上回った．しかし，BLEU での評価と同じく，ステップ数の増加はスコア向上に寄与しない．図 6.3 はコメント配信時刻とイベント発生時刻の時間差に関する分布である．まず，データセット中の人間が書いた市場コメントを，正規表現を用いることで定時刻のコメントか（*Regular* ）そうでないか（*Irregular*）に分類した．定時刻のコメントの場合，その事象の発生時刻とコメント配信時刻の時間差を 5 分単位で，例えば、0 = 0 分〜5 分のずれ、1 = 5 分〜10 分のずれで

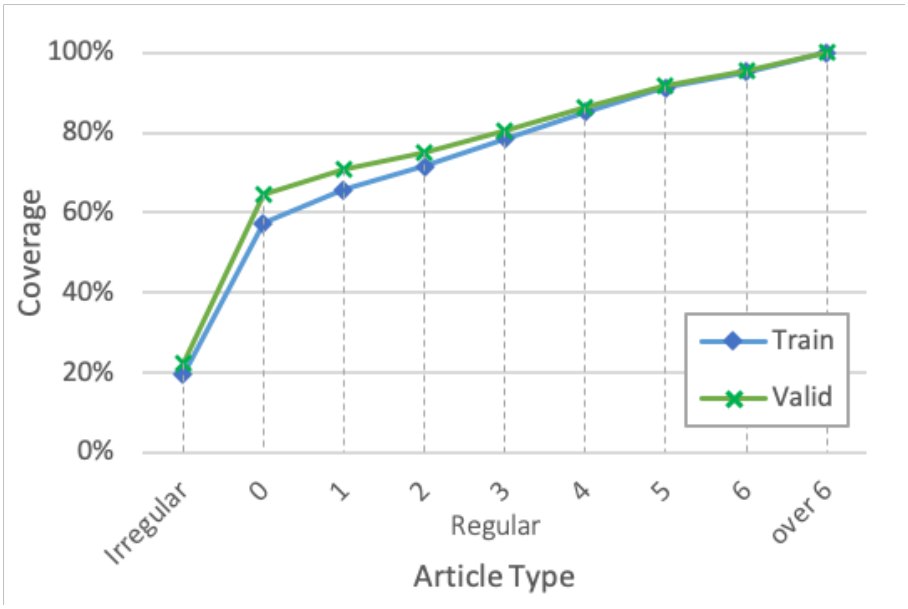



図 6.3: コメント配信時刻とイベント発生時刻の時間差に関するデータ分布.

分類した．つまり，図 6.3 では $n = 5$ を用いた場合、学習データで 89.05%、検証セットで 94.07%の時間差データ（図 6.3 の *Irregular* 値と 0 5 の合計）をカバーしていると示している．しかし，$\boldsymbol{x}_k$ を多く使用するとその分ノイズが多くなるため，トレードオフの関係である．表 6.3，表 6.4 および図 6.3 より，$n = 3$ または $n = 4$ を用いて，約 80% をカバーすることが，このデータセットでは最適な選択と考えられる．

表 6.5 に生成結果の例を示す．提案手法では，Gold のテキストと同じく，動向表現が最新の数値を使った場合と一致しない表現が生成できた．さらに，Copy 機構を用いた場合には，数値を正しく生成できている．一方，流暢さには問題が残っている．例えば，“$XX$ 円台後半” という表現の場合，$XX$ は 100，1,000 もしくは 10,000 などの丸め込んだ数値となるのが自然である．しかし生成された数値は丸め込んだ数値ではなかった．動向表現は Gold と同様の表現ができる一方，数値生成には問題が残っている．

表 6.4: 開発データでの動向一致・不一致評価 (%, P=precision, R=recall)

| | 一致 | | 不一致 | |
|---|---|---|---|---|
| | P | R | P | R |
| **Baseline** | | | | |
| | 98.68 | 98.78 | 27.57 | 24.72 |
| **Multi-timesteps** | | | | |
| $n = 0$ | 98.61 | 98.94 | 41.00 | 33.93 |
| 1 | 98.77 | 99.02 | 46.03 | 40.52 |
| 2 | 98.79 | 98.47 | 40.66 | 46.13 |
| 3 | 98.82 | 98.61 | 43.12 | 46.20 |
| 4 | 98.88 | 98.86 | 46.58 | 47.05 |
| 5 | 98.88 | 98.91 | 44.13 | 43.47 |
| 6 | 98.80 | 98.70 | 41.88 | 43.43 |

| | 一致 | | 不一致 | | 一致 | | 不一致 | |
|---|---|---|---|---|---|---|---|---|
| | P | R | P | R | P | R | P | R |
| | **+Attention** | | | | **+Copy** | | | |
| $n = 0$ | 98.90 | 99.08 | 51.35 | 45.28 | 98.85 | 99.25 | 53.90 | 42.61 |
| 1 | 98.76 | 99.28 | 56.83 | 41.91 | 98.81 | 99.02 | 45.86 | 39.80 |
| 2 | 98.99 | 98.78 | 45.00 | 49.74 | 98.71 | 99.06 | 47.08 | 38.56 |
| 3 | 98.93 | 98.87 | 44.65 | 46.25 | 98.73 | 98.95 | 45.38 | 38.79 |
| 4 | 98.80 | 98.44 | 30.60 | 36.29 | 98.68 | 98.68 | 36.46 | 36.06 |
| 5 | 98.69 | 98.14 | 27.88 | 34.92 | 98.57 | 98.57 | 34.04 | 31.54 |
| 6 | 98.70 | 97.11 | 25.54 | 38.74 | 98.46 | 98.81 | 32.35 | 26.17 |

表 6.5: 直近の 5 分足と前日の終値との差が-48.91 円の生成テキスト例.

| モデル | 生成テキスト | 動向との一致 |
|---|---|---|
| 正解文 | 日経平均、続伸で始まる 9 円高の 17024 円 | × |
| ベースライン | 日経平均、反落で始まる 16900 円台 | ✓ |
| **Multi-timesteps** | | |
| $n = 3$ | 日経平均、小幅続伸で始まる | × |
| +Attention | 日経平均、小幅続伸で始まる | × |
| +Copy | 日経平均、小幅続伸で始まる　17024円台後半 | × |

## 6.5 予測不可能な属性

Data-to-text のデータセットは実世界で得られたデータとテキストの組み合わせによって作られるものと，実世界を模して，つまり擬似的に入出力を想定してデータやテキストを獲得して作られるものがある．実世界を模して得られたデータセットとは，例えばクラウドソーシングによってデータやテキストを獲得して作成されたデータセットである．このようなデータセットは，入出力をコントロールして作成されている．例えば，E2E チャレンジデータセット (Novikova et al., 2017) の出力テキストは入力の Meaning Representation(MR) をすべて言及するように作られている．また ToTTo (Parikh et al., 2020) は wikipedia のテーブルをハイライト付きで与えて，1 文章を生成するタスクのためのデータセットで，入力となるデータを目的のテキストに合わせて作成してるデータセットと言える．

一方，実世界で得られたデータとテキストで作られたデータセットの場合，データやテキストをそのまま利用するため，目的のテキストに合わせて入力データをコントロールできない．そのため，入力データから出力テキストを完全に推定できないことがある．Rotowire は実データを用いた data-to-text のデータセットの一つである．Rotowire データセットを提案した Wiseman et al. (2017) では，データをそのまま利用していたが，その後の研究 (Saleh et al., 2019; Iso et al., 2019; Gong et al., 2019) では，出力テキスト生成をコントロールするために，入力データにはない予測不可能な属性である筆者情報を用いることで，content selsction や correct order の精度が高くなり，生成精度が向上している．このように，実データを用いた data-to-text では入力データからは予測できない属性に関する情報を追加することで精度向上が見られる．

日経平均株価の概況テキストは時系列データに対して逐次作成されるテキストであり，6.4 節に示したように，noisy なデータセットである．一方，実データを用いたデータセットであるため，より実用に近い課題のデータセットである．逐次作成される市況データコメントは，同じ市況データを見ていても媒体によって書き方が違っていたり，市況の状態に応じて作成される文章か，決まった時刻に作成される定時文章か，そして，今の状況のみを反映して作成される速報か，今後の状況まで反映して作成される文章かによって，言及内容や文体が変わる．専門家はそれらを反映した文章を書いており，投資家もそれが反映された文章を利用する．つまり，データセットのテキストは媒体やシチュエーションを反映して書かれた文章である．そこで，先行研究で用いられているデータセットの入力データと出力テキストとの関係性を分析し，テキストの推定に必要な追加入力ラベルを提案する．また，追加入力ラベルの追加タイミングについても再検討し、妥当な追加タイミングについて提案する．

### 6.5.1 データセット分析

ここでは先行研究 (Murakami et al., 2017; Aoki et al., 2018) で用いられたデータセットについて，データセットの作成手順を説明し，それによって発生する問題を挙げ，分析し，解決策を探る．

#### データセット作成手法

Murakami et al. (2017) で作成したデータセットは日経平均株価と，日経平均株価について言及しているニュース記事のヘッドラインを使用した．日経平均株価は非常に短い時間間隔（15 秒）で変動しているが，この数値データをそのまま用いると非常に多いデータ量となるため，一般的にはより大きい間隔で区切って使用する．今回使用したデータは 5 分間隔に区切ったデータ，つまり 5 分足データを用いる．この時，各間隔の代表値は区切った時刻の最後の値を用い，その代表値からなる新しい時系列データが作成される．なお，このような間隔分割処理は，入力データが連続的である場合には珍しくなく，このデータセット固有の問題ではない．このように作成した時系列データを用いて，概況テキストと組み合わせてデータセットを作成する．6.4 節で述べた通り，概況テキストにはその記事が配信された時刻が付与されており，その時刻を基準に入力データとテキストを組み合わせている．そのため，本データセットでは，ある 5 分間に複数の概況テキストが配信された場合には，それらの入力データは同じになる．実際，検証用データセットにはり合計 1,751 件の入出力データセットのペアが存在するが，入力データの種類は 1,176 種類にとどまり，1 対多のアライメントが存在していることがわかる．また，データセットに含まれる概況テキストには，日経平均，もしくは，東証で始まるテキストの 2 種類があることやある程度決まった時刻（定時刻）に配信されるテキストが存在することも分かっている．

#### 同じ入力データで整列した複数の概況テキストの比較

表 6.6 に，同じ入力での概況テキストの例を示す．定時刻に配信されたある概況テキストを基準（Pivot comment）とし，同一入力データの他の概況テキストと比較する．その結果，他の概況テキストは少なくとも次の 4 つの点で Pivot comment と異なっていることが分かる：

**Content order:** 概況テキスト中の内容表現の順序が異なる．

**Lexical choice:** 内容は同じだが，表現方法（単語）が異なる．

**Informativeness:** 情報量が異なる．

**Others:** 上記以外

表 6.6 は一例だが，表に記載されている違いは一般的なものである．

表 6.6: 同じ入力データで異なる概況テキストの例.

| 違いの種類 | 概況テキスト例 |
|---|---|
| (Pivot comment) | 日経平均、続伸で始まる。9 円高の 17024 円 |
| Content order | 日経平均、9 円高の 17024 円で続伸して始まる。 |
| Lexical choice | 日経平均、続伸で始まる。始値は 9 円高の 17024 円 |
| Informativeness | 日経平均、続伸で始まる。 |
| Others | 日経平均、上げ幅 100 円 超える。 |

### 差異を生む要因

対象のデータに含まれる概況テキストの中には，9 時，10 時，11 時 30 分，12 時 30 分，14 時，15 時などある一定の時間帯の市場の状況を反映しているものがある．これは *regular reports* である．一方，それ以外の概況テキストは，報告すべき事象が発生したときに発信される *prompt reports* である．また，今回のデータセットには，「日経平均」で始まるテキストと，「東証」から始まるテキストの 2 つの異なる書き方がある．検証用データセットでは，62.70%の概況テキストが「日経平均」で始まり，その他は「東証」で始まる．さらに，あるテキストは特別なトークンである ♦ から始まっており，このテキストが重要なテキストであることが示されている[5]．これら 3 つの属性を *regular/prompt*， *writing style*，および *supposed-importance* とする．次節以降，これらの属性が概況テキストにどのような影響を与えるかを検証する．

**情報量の分析**　概況テキストの長さは，概況テキストに含まれる情報量と相関があると考え，概況テキストの平均的な長さを分析した．その結果，「regular」概況テキストの平均トークン数は 8.04，「prompt」概況テキストの平均トークン数は 6.14 となり，「regular」概況テキストの方がより詳細な情報を含んでいることが分かった．これは，通常の概況テキストの方がより詳細な情報を含んでいることを意味する．同様に，♦ から始まる “supposed-importance” 概況テキストの平均トークン数は 8.21 であり，そうでない概況テキストの平均トークン数は 7.24 であった。このことから，“supposed-importance” 概況テキストの方が情報量が多いことが分かる。また， *writing style* は，「日経平均」で始まる概況テキストの平均トークン数が 7.96 であったのに対し，「東証」で始まる概況テキストの平均トークン数は平均 6.99 個だった．このことから，「日経平均」で始まる概況テキストの方が，「東証」で始まる概況テキストよりも情報量が多いことが分かる．

[5]前処理時に削除されるため，生成は行われない．

**内容と順序** 時間表現を「TIME」，株価表現を「PRICE」，株価変動表現を「MOVE」とし，概況テキスト中の内容（TIME，PRICE，MOVE）とその順序を検証した．例えば，「日経平均、続伸で始まる。17024 円の 9 円高」 は，[TIME，MOVE，PRICE，PRICE，MOVE] と置き換えられる．

「regular」概況テキストでは, [MOVE，TIME，PRICE，MOVE，PRICE] が最も多く (17.23%), 次いで [TIME，MOVE] が多かった (16.34%)．一方,「prompt」テキストでは，[MOVE，PRICE] が最も多かった (56.54%)．「regular」テキストには TIME が含まれるが，臨時テキストには含まれない．“supposed-importance” 概況テキストには，[MOVE，TIME，PRICE，MOVE，PRICE] が最も多く 35.65%を占めている．そうでないテキストには [MOVE，TIME，PRICE，MOVE，PRICE] が 1,103 のうち 1 文に留まった一方, [TIME，MOVE] が最も多かった (19.95%). regular/prompt テキストや，“supposed-importance” テキストかそうでないかの概況テキストの違いは，TIME の有無や位置が異なると分かる．

*writing style* に着目すると,「日経平均」で始まる 21.22%の概況テキストは [MOVE，TIME，PRICE，MOVE，PRICE] となり，次いで 20.77%のテキストが [MOVE，PRICE] であった．一方,「東証」で始まる概況テキストにはこの両方がなく，最も多い内容とその順序は [TIME，MOVE] で 28.79%を占めている．

**語彙選択の分析** 語彙選択の違いについて検討した．表 6.6 の 3 番目の例では，”始値”という表現が含まれているが，これは Pivot コメントには含まれていない．この 2 つのコメントのような語彙選択の違いは，入力データからは捉えられないことが多く，予測できない属性に依存することがある．

**属性予測による分析** 最後に，これらの属性が市場価格データから予測可能かどうかを検討した．表 6.7 にその結果を示す．

表 6.7: 属性予測（%）．ベースラインは各属性の多数派がデータセットで占める割合

| 属性 | ベースライン | 精度 |
|---|---|---|
| *regular/prompt* | 76.87 | 78.62 |
| *writing style* | 62.70 | 73.39 |
| *supposed-importance* | 62.99 | 65.87 |

分類には， 6.5.3 の生成モデルの encoder として Murakami et al. (2017) が使用した encoder と softmax 層からなる分類器を使用した．「regular」テキストがデータセット中で占める割合が 76.87%（残りが「prompt」テキスト）であり，本実験の属性予測において，regular/prompt の分類精度は 78.62%であった．regular/prompt 分類器が多数派のベース

ラインより優れていないことを示唆する．また，“supposed-importance” 概況テキストの分類予測精度は 65.87%であり，“supposed-importance” でない概況テキストがデータセット内で占める割合の 62.99%から大きな差はない．一方,*writing style* の属性予測の精度は 73.39%で，大多数を占める「日経平均」で始まる概況テキストの割合 62.70%を上回った．しかし，これは「東証」から始まる概況テキストは，その全てが「regular」テキストであり，時間と強く関連しているためと考えられる．「regular」テキストが参照する各時刻から 10 分後までの範囲に絞って属性予測を行ったところ，精度は 65.96%であり，ベースラインと大きな差はなかった．

これらの結果から，属性はほぼ予測不可能であることが示唆され，今回の 3 つの属性に由来する概況テキストの差異は，入力された株価の系列からは捉えることができない．

### 6.5.2 ラベルを用いた概況テキスト生成

6.5.1 項での分析結果を受け，本項では実験の準備として予測不可能な属性の情報をどのように概況テキスト生成モデルに統合するかを説明する．まず，予測できない属性を表すラベルとして，*writing style* と *importance* を定義する．そして，これらのラベルを追加情報としたモデルについて説明する．なお，本実験の目的は，新しい概況テキスト生成モデルを提案することではなく，予測不可能な属性の影響を示し，そのような予測不可能な属性は入力の一部として与えられるべきであることを主張することである．

#### 分類ラベルの設計

6.5.1 項では, 概況テキストの違いには,「regular」テキストか「prompt」テキストか,「日経平均」から始まるか「東証」から始まるか, 元のテキストが ♦ から始まっている “supposedly important” のテキストかどうかなどがあることを指摘した．このように概況テキストの差異要因である，予測できない属性を反映させて作成したラベルを表 6.8 に示す．Murakami et al. (2017); Aoki et al. (2018) のモデルで導入されている各時間の *time* ラベルに加え，新たに *writing style* と *importance* の 2 つのラベルを作成した．

*writing style* ラベルは，6.5.1 項で分析した文体属性のを示しており，日経平均と，東証の 2 つのうちどちらかの値を持つ．

*importance* ラベルは新たに定義するラベルで,「regular」テキストか「prompt」テキストかの属性と，“supposedly important” テキストかどうかの属性の 2 種類の属性を組み合わせたラベルである．つまり，regular ∩ supposedly-important, prompt ∩ supposedly-important, regular ∩ not supposedly-important, and prompt ∩ not supposedly-important の 4 つのうちのいずれかを持つラベルである．

表 6.8: 分類ラベルとその分類値

| 分類ラベル名 | 分類値 |
|---|---|
| *writing style* | 日経平均, 東証 |
| *importance* | regular ∩ supposedly-important,<br>prompt ∩ supposedly-important,<br>regular ∩ not supposedly-important,<br>prompt ∩ not supposedly important |
| *time* | 9 時台（9:00-9:59am），10 時台（10:00-10:59am）, . . .<br>Murakami et al. (2017) によるラベル. |

### モデル

前項で設定したラベルをモデルに追加する場合，その追加箇所を 図 6.4 のように(a) 入力データと同時に追加, (b) Encoder の後に追加（Decoder の直前に追加）, および(c) Decoder の各ステップに追加の 3 通り検討する．

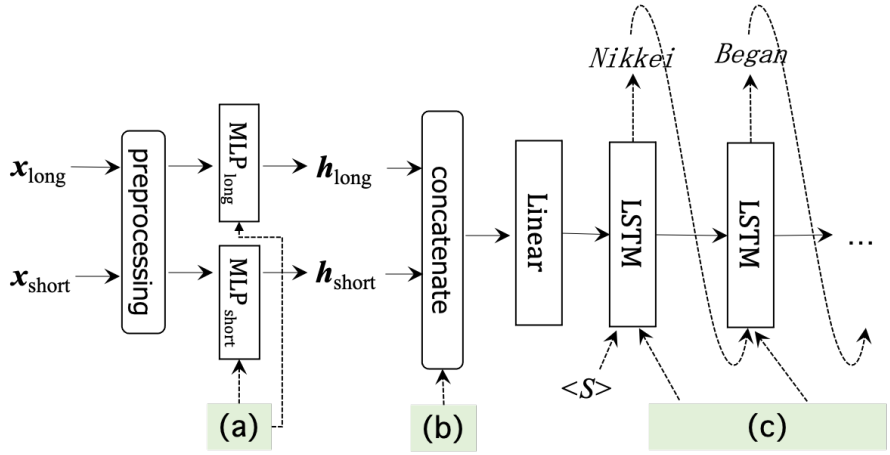



図 6.4: 属性ラベルを用いた概況テキスト生成モデルの概要

ラベルの各値は，学習によって得られる埋め込みとして表現される．これを単にラベル埋め込みと呼び，$\boldsymbol{l}$ と表記する．ネットワークの状態を計算する式を Murakami et al. (2017) のモデルから次のように変更する．(a) の場合，式 (6.4) と (6.5) を以下のように変更する．

$$\boldsymbol{h}_{\text{long}} = \text{MLP}[\boldsymbol{x}'_{\text{long}}; \boldsymbol{l}], \tag{6.19}$$

$$\boldsymbol{h}_{\text{short}} = \text{MLP}[\boldsymbol{x}'_{\text{short}}; \boldsymbol{l}]. \tag{6.20}$$

(b) の場合, 式 (6.6) を以下のように変更する.

$$\boldsymbol{m} = \boldsymbol{W}_m[\boldsymbol{h}_{\text{long}}; \boldsymbol{h}_{\text{short}}; \boldsymbol{l}] + \boldsymbol{b}_m. \tag{6.21}$$

(c) の場合，式 (6.7) を以下のように変更する.

$$\boldsymbol{s}_i = \text{LSTM}([\boldsymbol{w}_{i-1}; \boldsymbol{l}], \boldsymbol{s}_{i-1}). \tag{6.22}$$

1 つ以上のラベルを用いる場合，ラベルの埋め込みベクトルを結合して用いる．例えば，3 つのラベルを 1 度に使う場合， $\boldsymbol{l}$ は以下となる:

$$\boldsymbol{l} = [\boldsymbol{l}_{\text{time}}; \boldsymbol{l}_{\text{writing style}}; \boldsymbol{l}_{\text{importance}}] \tag{6.23}$$

このとき，$\boldsymbol{l}_{\text{time}}$ は *time* ラベル埋め込み，$\boldsymbol{l}_{\text{writing style}}$ は *writing style* ラベル埋め込み，$\boldsymbol{l}_{\text{importance}}$ は *importance* ラベル埋め込みである.

### 6.5.3 実験設定

前項で設定した通り，(1) *time*, (2) *writing style*, および(3) *importance*. の 3 つのラベルを使用し，そのラベルの追加箇所を(a) 入力データと同時に追加, (b) Encoder の後に追加（Decoder の直前に追加）および(c) Decoder の各ステップに追加，の 3 通りで実験を行う.

#### データセット

実験には，時系列データとして，Thomson Reuters DataScope Select[6]から，2011 年 12 月から 2016 年 9 月までの期間の日経平均株価指数の時系列データを収集し，実験に使用した．概況テキストとして，日経 QUICK ニュース社が提供している日経平均株価に言及しているニュース記事のヘッドラインを使用した．なお，米国株式や外国為替などの日経平均株価以外の記述は取り除いた．また Murakami et al. (2017) と同様に，価格に関する単語を演算タグに変換した．2011 年 12 月から 2015 年 9 月の期間のデータを学習データ（12,391 件），2015 年 10 月から 2016 年 3 月の期間のデータを開発データ（1,751 件），2016 年 4 月から 2016 年 9 月の期間のデータを評価データ（1,689 件）として使用した.

#### ハイパーパラメータ

先行研究と同様，次の設定で実験を行う．各 MLP の隠れ状態の次元は 256，Decoder の LSTM の隠れ状態の次元は 256，短期と長期のベクトルの長さは，$\boldsymbol{x}_{long}$ は $M = 7$，$\boldsymbol{x}_{short}$

[6] https://hosted.datascope.reuters.com/DataScope/

は $N = 62$ とする．単語埋め込みベクトルは 128，3 つのラベル *time*, *writing style*, および *importance* の埋め込みベクトルの次元は 64 とする．ミニバッチサイズは 100，エポック数は 150 とし，各パラメータの最適化手法には Adam (Kingma and Ba, 2015) を使用し，学習率は $1 \times 10^{-4}$ とした．学習時に開発データで計算された BLEU が連続して下がった場合には学習を終了し，各エポックのモデルの中で，開発データに対する BLEU が最大となったモデルを使用して，評価を行う．

#### 評価指標

生成文の評価は，正解文との一致を評価可能な BLEU(Papineni et al., 2002) で行った．また，正解文章と比較し，動向表現[7]が正しく出現できているかを評価した．

### 6.5.4 実験結果

表 6.9: BLEU (%)

| | (a) | (b) | (c) |
|---|---|---|---|
| 開発データ | ラベルなし: 35.82 | | |
| *time* label | 37.42 | 40.78 | 40.44 |
| *writing style* label | 43.51 | 43.52 | 44.28 |
| *importance* label | 48.78 | 48.50 | **50.18** |
| 評価データ | ラベルなし: 36.79 | | |
| *time* label | 40.42 | 41.40 | 39.66 |
| *writing style* label | 46.69 | 46.54 | 47.36 |
| *importance* label | **52.77** | 51.90 | 52.15 |

BLEU 評価結果を表 6.9 に示す．ラベルありの場合に，最もスコアが低かったのは (a) の *time* ラベルで，Murakami et al. (2017); Aoki et al. (2018) の先行研究で使用した *time* ラベルはラベルなしと比べてスコアの向上に貢献したものの，最も効果的なラベルとは言えなかった．データセット解析を経て作成した *writing style* ラベルと *importance* ラベルの使用は，スコアの向上に有意に効果があることが分かった．開発データでの 1 つのラベルを使った場合の最高 BLEU スコアは，(c) を *importance* ラベルを追加した場合の 50.18% で，ラベルなしの場合よりも 15%向上し，また (c) で *time* ラベルを用いた先行研究の設定値よりも 10%以上高い値を示している．評価データでは，(a) で *important* ラベルを追加し

[7]続伸，反発，上げ，続落，反落，下げ

表 6.10: 開発データセットによる動作表現による評価 (%)

| | (a) | (b) | (c) |
|---|---|---|---|
| **精度** | ラベルなし: 89.27 | | |
| *time* label | 90.26 | 90.88 | 90.96 |
| *writing style* label | 90.61 | 90.64 | 90.68 |
| *importance* label | **91.40** | 91.27 | **91.40** |
| **特異度** | ラベルなし: 93.25 | | |
| *time* label | 93.86 | 94.05 | 94.08 |
| *writing style* label | 94.06 | 93.97 | 94.04 |
| *importance* label | 95.27 | 95.08 | **95.29** |
| **F 値** | ラベルなし: 56.96 | | |
| *time* label | 60.91 | 63.82 | 64.68 |
| *writing style* label | 63.17 | 63.36 | 63.83 |
| *importance* label | 65.01 | 65.08 | **65.19** |

た場合に最も高いスコアが得られ，最も低かったのは先行研究の設定の (c) で *time* ラベルを追加した場合だった．

表 6.10 に動向表現生成精度の評価結果を示す．ラベルを追加することで，文章生成精度が上がるだけでなく，正しい動向表現を選べるようになっていることが分かる．BLEU と同じく，ラベルを追加しない場合と比較して，全ての場合でラベルを追加した場合に精度が高くなった．しかも，精度だけでなく特異度や F 値も向上していることより，出すべき時に正しく生成できていることがわかる，特に，(c) で *importance* ラベルを追加すると，精度だけでなく特異度や F 値も最も高くなった．

以上より，ラベル自体には，データの動向を表す要素は含まれていないにもかかわらず，文体として正しい文章が生成できるだけでなく，データを正しく記述することにも貢献している．

同一入力データで生成された文章を表 6.11 に示す．ラベルを全く入れない場合は生成コメントが変わらなかった一方，正解のテキストと同一のラベルを各々与えた場合では，全て異なるコメントが生成できており，さらに各々正解テキストと同じようなコメントであることがわかる．

表 6.11: 同一入力データでのラベル追加なし・ありの生成テキスト例

| | |
|---|---|
| 正解文 | |
| (I) | 日経平均、続落 大引けは 69 円安の 17697 円 |
| (Ⅱ) | 日経平均大引け、続落 |
| (Ⅲ) | 東証大引け、続落 |
| 生成結果 | |
| ラベルなし | |
| | 日経平均、続落 大引けは 69 円安の 17697 円 |
| 全てのラベルあり | |
| (I) | 日経平均、続落 大引けは 69 円安 |
| (Ⅱ) | 日経平均大引け、続落 |
| (Ⅲ) | 東証大引け、続落 |

## 6.6 まとめと今後の課題

本章では，data-to-text の時系列データと動向説明文の対応関係獲得課題を題材に問題を２つ挙げ，分析し，解決策を探った．

１つ目の問題は，時系列データを用いる課題で発生する特有の問題である．これは，マルチモーダル情報の変換における適切な入出力の表現方法に関する問題とも言える．時系列データとその時系列データの内容を逐次解説するテキストでデータセットを構築する場合，テキストの発表時刻を基に時系列データを取得する手法が主に用いられているが，データのイベント発生時刻とテキスト発表時刻の不整合が起こりうる．そこで，テキスト発表時刻はイベント発生時刻よりも遅れることに注目し，テキスト発表時刻以前のデータに対しても参照できる枠組みを提案した．この解決策により，時系列データを用いた data-to-text における時刻の不整合の問題に対応できた．この手法は，入力の枠組み以外は先行研究の枠組みを大きく変えず，簡単に導入可能である特徴がある．

２つ目の問題は，データセットにおける予測不可能な属性の問題，つまり，入力から出力の属性が予測できない問題である．これは，マルチモーダル情報の変換における Generative approaches による，入力に対応する正解が複数存在しうる問題と関係が深い．時系列データとその時系列データの内容を逐次解説するテキストで構築されたデータセットでは，同一データであるが，異なるテキストとなっているデータペアが存在した．このようにデータセットには予測不可能な属性が存在し，このデータセットを使った場合入力から正しいテキストを生成することができないことが判明した．データセットを分析して得られた予測不可能な属性から 2 つのラベルを作成し，ニューラルネットワークベースモデルに新たな入力として追加したところ，ラベルの使用によりテキストの生成性能が向上することはもちろんのこと，ラベルの使用によりデータの内容をより正しく生成できる副次的な効果も確認された．テキスト生成モデルを予測不可能な属性が存在する環境で学習させるのではなく，予測不可能な属性を入力の一部として与えるべきであると示唆している．

以上，2 つの問題の解決策は，実データを用いたモダリティ間の関係性獲得には汎用的な手法を直接使えるわけではなく，モダリティによってある程度の調整が必須であることを示唆している．特に，モダリティ変換を行いたい方向性がある場合，例えば専門性の高いデータや文章を扱う場合には，その方向性を制御する必要がある．

今後の課題は，入出力データの表現方法の自動獲得や，モダリティ変換の方向性の自動獲得である．本章で扱ったデータセットには人が気付ける特徴があったため，人手による入出力の分析により，適切な表現方法や，モダリティ変換の方向性を得ることができた．一方暗黙知など，人が気付けない特徴も存在しうるため，それらにより入出力データが特徴付けられていたり変換の方向性が決まる場合には，人手による分析では適切な表現方法や，モダリティ変換の方向性を得ることができない．そのため，人手を介さないこれらの自動獲得手法は，モダリティ変換を行う上で必須となると考えている．

# 第7章　結論

本論文では，言語と非言語の対応関係を捉えるタスクとして，マルチモーダル情報のモダリティ変換を行う生成課題に取り組んだ．マルチモーダル情報の生成による変換手法では，出力データの両方のモダリティを捉えた上で，モデルを構築する必要がある．そのため，モダリティ変換はモダリティ毎にそれぞれモデルを作る必要がある．本論文では，言語を中核に据えて，非言語とのモダリティ変換を 3 つ扱い，言語の特性を捉えたモダリティ変換課題に取り組んだ．

はじめに，言語から非言語の生成課題として，自然言語理解によるロボット動作生成課題を扱い，その手法の提案を行った．この課題では，人間とロボットが共存する場面を想定したため，まず，ロボットが人の動作を真似て行うことができるように，基本動作を組み合わせてひとつの行為を生成する枠組みの時系列 AAM を構築し，複数の基本動作から複雑な行為を表現する対応関係を明確にした．従来のロボット動作構成は，関節角を直接指示して動作を生成する必要があったため，可読性が低かったが，時系列 AAM は，基本動作を組合せて動作構成するため，可読性が高く，人が新たな動作をロボットに指示する場合にも容易である．また，作成動作を新たな基本動作として定義でき，複雑な動作生成へ拡張性がある．ロボットの動作構成が可能になった上で，言葉とロボット動作の対応関係を捉える手法として，ニューラルネットワークを基にしたモデルにより，分散表現と時系列 AAM による動作表現を対応付ける手法を提案し，有効性を確認した．特に，分散表現生成に大きなコーパスを用いることで，複雑な言葉と動作への対応が可能なことを確認した．さらに，全く新しい言葉の動作でも学習した言葉の動作から言葉の動作を推測して動作を行えることを確認した．複数の動作生成に適用できるニューラルネットワークの構造では，単純な構造ではなく，言葉の特徴を考慮した枠組みで高い精度を達成した．このことは，異なるモダリティ間に直接的なパターンの対応関係はないことを示唆しており，対応関係を捉えて，適切に生成を行うためには，入力特徴を考慮した枠組みにする必要があると言える．またこの枠組みにより，言葉の曖昧さを動作というモダリティに限定して対応関係を獲得することで，モダリティ変換することができた．

次に，非言語から言語の生成課題として，一般ドメイン実況生成と概況テキスト生成を扱った．これは新たな課題の提案であり，既存のクローズドメインの実況生成では利用できていた分野特有のデータを使えず，映像のみからテキストを生成する必要がある課題設定である．一方で，ドメインを限らない課題とすることで，同じ状況であっても，複数の表現方法が考えられる，言語の広がりを許容する新たな課題の提案となった．動画中のイベ

ントを抽出してテキストで説明する Dense captioning とも異なり，時間的位置，内容，ビデオ内のオブジェクトの相関が，ビデオ内のイベントと部分的にしか整合していない．まず，様々なドメインの人間の動作を含むビデオに対して実況を収録し，新しい大規模データセットの構築を行い，24,000 件以上の実況データを得た．その上で，実況生成に必要なタイミング推定と発話生成に取り組んだ．タイミング推定では，トリミングされていないビデオシーケンス中の人間の活動を推定できる TAL を用い，発話生成では，テキスト生成で有用性が示されている Transformer モデルを用いた．発話生成において，映像のみを入力とする場合と前時刻の実況も含めて入力とする場合を比較すると，前時刻の実況も含めて入力した場合に精度が向上した．実況が動画内のイベントと部分的にしか一致していないという性質を持っており，動画の内容と一致しない実況を生成する必要があるときに，前時刻の実況がモデルの決定を助け，なお，既存のクローズドメインデータセットにおいて，分野特有のデータを用いた場合には明らかに精度が向上した．これは，映像には存在しない，ドメイン内情報へのアクセスの重要性を示しているが，映像にない情報へのアクセスは，クローズドメインのみに重要なわけではなく，一般ドメインにおいても重要な点である．例えば，ダーツの動画に対する実況を行う場合，矢がダーツボード，特にその中心部に当たったかについて言及しうる．これはダーツという遊びの目的は，矢をダーツボード (特にその中心部) に当てることであるからであるが，映像のみからなる訓練データのみから，ダーツのルールを把握することは容易ではない．このように，今後は一般ドメインにおいても，なんらかの外部知識の利用が必要となると考えられる．

最後に，実世界で得られたデータやテキストを用いて対応間関係を得ようとする場合に発生する問題に取り組んだ．これは，限られたデータから適切に目的のテキストを生成する手法の獲得であり，複数の解釈が可能なデータから，言葉によって解釈の方向性を決定する課題とも言える．特に，data-to-text の時系列データと動向説明文の対応関係獲得課題を題材に問題を 2 点上げ，分析し，解決策を探った．1 つ目の問題は，時系列データを用いる課題で発生する特有の問題である．時系列データとその時系列データの内容を逐次解説するテキストでデータセットを構築する場合，テキストの発表時刻を基に時系列データを取得する手法が主に用いられているが，データのイベント発生時刻とテキスト発表時刻の不整合が起こりうる．そこで，テキスト発表時刻はイベント発生時刻よりも遅れることに注目し，テキスト発表時刻以前のデータに対しても参照できる枠組みを提案した．この解決策により，時系列データを用いた data-to-text における時刻の不整合の問題に対応できた．この手法は，入力の枠組み以外は先行研究の枠組みを大きく変えず，簡単に導入可能である特徴がある．2 つ目の問題は，データセットにおける予測不可能な属性の問題，つまり，入力から出力の属性が予測できない問題である．時系列データとその時系列データの内容を逐次解説するテキストで構築されたデータセットでは，同一データであるが，異なるテキストとなっているデータペアが存在した．このようにデータセットには予測不可能な属性が存在し，このデータセットを使った場合入力から正しいテキストを生成することができないことが判明した．データセットを分析して得られた予測不可能な属性から 2 つのラベルを作成し，ニューラルネットワークベースモデルに新たな入力として追加したところ，

ラベルの使用によりテキストの生成性能が向上することはもちろんのこと，ラベルの使用によりデータの内容をより正しく生成できる副次的な効果も確認された．テキスト生成モデルを予測不可能な属性が存在する環境で学習させるのではなく，予測不可能な属性を入力の一部として与えるべきであると示唆している．

以上より，本論文では，実用に向けた言語と非言語のマルチモーダル情報のモダリティ変換を行うことで，言語とロボット動作，言語と一般ドメイン動画，言語と時系列数値データの各モダリティ変換手法を得ることができた．これは，各入出力データのモダリティに適した表現方法を用いたことによる貢献も大きい．さらに，言葉と特定の非言語間のモダリティ変換を扱うことで，言語の曖昧さの許容や広がり，解釈の可能性を特定のモダリティを介して表すことができた．また，実世界で言語と非言語の対応関係を捉える課題を行うことで，異なるモダリティの関係性を正しく捉えるには，入出力データそれぞれの特徴を不足なく取得した上で，関係性を捉える適切な枠組みが必要であると示唆できた．本論文で扱った入出力データのモダリティやそれらを用いたモダリティ変換は限られた範囲であるため，これを直接異なるモダリティに用いることはできない．しかし，似たような特徴を持つモダリティ変換の手掛かりとなると考えている．本研究により，異なるモダリティの関係性を正しく捉えるには，入出力データそれぞれの特徴を捉える必要があるとしたが，データには暗黙知など，人が気付けない特徴も存在しうる．人が気付けない特徴により，入出力データが特徴付けられていたり，モダリティ変換の方向性が決まる場合には，人手による分析では適切な表現方法や，モダリティ変換の方向性を得ることができない．そのため，人手を介さない入出力データの表現方法の自動獲得手法や，モダリティ変換の方向性の自動獲得手法は，今後マルチモーダル情報のモダリティ変換を行う上で必須となると考えている．

# 謝辞

本研究を進めるにあたり，多くの方々にご協力を頂きました．心より感謝致します．

まず，本論文を執筆するに辺り，ご指導下さいましたお茶の水女子大学人間文化創成科学研究科の小林一郎教授に深謝いたします．

また，副査を担当くださいました本研究科人間文化創成科学研究科小口正人教授，戸次大介准教授，伊藤貴之教授，五十嵐悠紀准教授，にも感謝いたします．終始適切な助言を賜り，ご指導頂いた，統計数理研究所数理・推論研究系の持橋大地准教授，産業技術総合研究所人工知能センターの麻生英樹氏，高村大也氏，東京大学大学院情報理工学系研究科の宮尾祐介教授，大阪大学基礎工学研究科の長井隆行教授，電気通信大学知能機械工学専攻の中村友昭准教授に感謝致します．

また鋭いご指摘を頂いたり，議論を交わす機会を与えてくださった産業技術総合研究所人工知能センターの石垣達也氏，Edison Marrese-Taylor 氏，上原由衣氏，能地宏氏，Goran Topić 氏に感謝いたします．

本研究の実験においてはクックパッド株式会社よりデータを提供していただきました．ここに感謝の意を表します．

最後に，日々の研究室生活を豊かなものとしてくれた研究室のメンバー，友人の皆様に心から感謝の意を表します．

# 参考文献

2016. Can neural machine translation do simultaneous translation? **arXiv:1606.02012 [cs]**.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic Propositional Image Caption Evaluation. In **Computer Vision – ECCV 2016**, Lecture Notes in Computer Science, pages 382–398, Cham. Springer International Publishing.

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In **Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing**, pages 502–512, Cambridge, MA. Association for Computational Linguistics.

Tatsuya Aoki, Akira Miyazawa, Tatsuya Ishigaki, Keiichi Goshima, Kasumi Aoki, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2018. Generating market comments referring to external resources. In **Proceedings of the 11th International Conference on Natural Language Generation**, pages 135–139, Tilburg University, The Netherlands. Association for Computational Linguistics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In **Proceedings of the 12th Language Resources and Evaluation Conference**, pages 4218–4222, Marseille, France. European Language Resources Association.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. 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**.

Christopher Baldassano, Janice Chen, Asieh Zadbood, Jonathan W Pillow, Uri Hasson, and Kenneth A Norman. 2017. Discovering event structure in continuous narrative perception and memory. **Neuron**, 95(3):709–721.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2019. Multimodal machine learning: A survey and taxonomy. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 41(2):423–443.

Anja Belz. 2007. Probabilistic generation of weather forecast texts. In **Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference**, pages 164–171, Rochester, New York. Association for Computational Linguistics.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. **Natural Language Engineering**, 14(4):431–455.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. **Journal of machine learning research**, 3(Feb):1137–1155.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, 5(2):157–166.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. **Behavior research methods**, 39(3):510–526.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In **proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, pages 6299–6308.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TExt Representation Learning. In **Computer Vision – ECCV 2020**, Lecture Notes in Computer Science, pages 104–120, Cham. Springer International Publishing.

Heng-Tze Cheng, Feng-Tso Sun, Martin Griss, Paul Davis, Jianguo Li, and Di You. 2013. Nuactiv: Recognizing unseen new activities using semantic attribute-based learning. In **Proceeding of the 11th annual international conference on Mobile systems, applications, and services**, pages 361–374. ACM.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from Wikidata. In **Proceedings of the 15th Conference of the**

**European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers**, pages 633–642, Valencia, Spain. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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)**, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In **Proceedings of the 25th international conference on Machine learning**, pages 160–167. ACM.

Kees van Deemter, Mariet Theune, and Emiel Krahmer. 2005. Real versus template-based natural language generation: A false opposition? **Computational Linguistics**, 31:15–24.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. **Journal of the American society for information science**, 41(6):391.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)**, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. 2015. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, pages 961–970.

J. R. Firth. 1957. A synopsis of linguistic theory 1930-55. 1952-59:1–32.

Adam Gaier and David Ha. 2019. Weight agnostic neural networks. `https://weightagnostic.github.io`.

Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. 2017. Video captioning with attention-based lstm and semantic consistency. **IEEE Transactions on Multimedia**, 19(9).

F.A. Gers, J. Schmidhuber, and F. Cummins. 1999. Learning to forget: continual prediction with lstm. In **1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)**, volume 2, pages 850–855 vol.2.

Eli Goldberg, Norbert Driedger, and Richard I Kittredge. 1994. Using natural-language processing to produce weather forecasts. **IEEE Expert**, 9(2):45–53.

Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pages 3143–3152, Hong Kong, China. Association for Computational Linguistics.

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

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Zellig S Harris. 1954. Distributional structure. **Word**, 10(2-3):146–162.

Geoffrey E Hinton. 1984. Distributed representations.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. **Neural Computation**, 9(8):1735–1780.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. VLN BERT: A Recurrent Vision-and-Language BERT for Navigation. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**, pages 1643–1653.

Andrew J Hunt and Alan W Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In **1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings**, volume 1, pages 373–376. IEEE.

Tatsuya Ishigaki, Goran Topic, Yumi Hamazono, Hiroshi Noji, Ichiro Kobayashi, Yusuke Miyao, and Hiroya Takamura. 2021. Generating racing game commentary from vision,

language, and structured data. In **Proceedings of the 14th International Conference on Natural Language Generation**, pages 103–113, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2019. Learning to select, track, and generate for data-to-text. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pages 2102–2113, Florence, Italy. Association for Computational Linguistics.

Armand Joulin and Tomas Mikolov. 2015. Inferring algorithmic patterns with stack-augmented recurrent nets. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, **Advances in Neural Information Processing Systems 28**, pages 190–198. Curran Associates, Inc.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In **Proceedings of the 13th International Conference on Natural Language Generation**, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Byeong Jo Kim and Yong Suk Choi. 2020. Automatic baseball commentary generation using deep learning. In **Proceedings of the 35th Annual ACM Symposium on Applied Computing**, pages 1056–1065. Association for Computing Machinery, New York, NY, USA.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**.

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In **Advances in Neural Information Processing Systems**, pages 3581–3589.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. **arXiv preprint arXiv:1312.6114**.

Arne Köhn, Florian Stegen, and Timo Baumann. 2016. Mining the spoken Wikipedia for speech data and beyond. In **Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)**, pages 4644–4647, Portorož, Slovenia. European Language Resources Association (ELRA).

Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. 2002. Natural language description of human activities from video images based on concept hierarchy of actions. **International Journal of Computer Vision**, 50:171–184.

Ioannis Konstas and Mirella Lapata. 2012. Unsupervised concept-to-text generation with hypergraphs. In **Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pages 752–761, Montréal, Canada. Association for Computational Linguistics.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In **Proceedings of the IEEE International Conference on Computer Vision**, pages 706–715.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In **Advances in Neural Information Processing Systems**, volume 25. Curran Associates, Inc.

Karen Kukich. 1983. Design of a knowledge-based report generator. In **21st Annual Meeting of the Association for Computational Linguistics**, pages 145–150, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. 2013. Babytalk: Understanding and generating simple image descriptions. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 35(12):2891–2903.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pages 7871–7880, Online. Association for Computational Linguistics.

Liunian Li and Xiaojun Wan. 2018. Point precisely: Towards ensuring the precision of data in generated texts using delayed copy mechanism. In **Proceedings of the 27th International Conference on Computational Linguistics**, pages 1044–1055, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In **Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP**, pages 91–99, Suntec, Singapore. Association for Computational Linguistics.

Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. 2020. Fast Learning of Temporal Action Proposal via Dense Boundary Generator. **Proceedings of the AAAI Conference on Artificial Intelligence**, 34(07):11499–11506.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018a. Table-to-text generation by structure-aware seq2seq learning. In **Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence**.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018b. Table-to-text generation by structure-aware seq2seq learning. **Proceedings of the AAAI Conference on Artificial Intelligence**, 32(1).

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pages 720–730, San Diego, California. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. **CoRR**, abs/1301.3781.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. In **NIPS Deep Learning Workshop**.

Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. 2019. Streamlined Dense Video Captioning. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**, pages 6588–6597.

Soichiro Murakami, Akihiko Watanabe, Akira Miyazawa, Keiichi Goshima, Toshihiko Yanase, Hiroya Takamura, and Yusuke Miyao. 2017. Learning to generate market comments from stock prices. In **Proceedings of the 55th Annual Meeting of the**

**Association for Computational Linguistics (Volume 1: Long Papers)**, pages 1374–1384, Vancouver, Canada. Association for Computational Linguistics.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In **Proceedings of the 27th International Conference on Machine Learning (ICML-10)**, pages 807–814. Omnipress.

Arvind Neelakantan, Quoc V. Le, and Ilya Sutskever. 2016. Neural programmer: Inducing latent programs with gradient descent. In **4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings**.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In **Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue**, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Tetsuya Ogata, Masamitsu Murase, Jun Tani, Kazunori Komatani, and Hiroshi G Okuno. 2007. Two-way translation of compound sentences and arm motions by recurrent neural networks. In **Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on**, pages 1858–1863. IEEE.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In **2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pages 5206–5210.

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

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pages 1173–1186, Online. Association for Computational Linguistics.

Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. 2019. Memory-attended recurrent network for video captioning. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**, pages 8347–8356.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 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)**, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG challenge: Neural models vs. templates. In **Proceedings of the 11th International Conference on Natural Language Generation**, pages 463–471, Tilburg University, The Netherlands. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. **Natural Language Engineering**, 3(1):57–87.

Cristian Rodriguez-Opazo, Edison Marrese-Taylor, Basura Fernando, Hongdong Li, and Stephen Gould. 2021. Dori: Discovering object relationships for moment localization of a natural language query in a video. In **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision**, pages 1079–1088.

Abdelrhman Saleh, Ramy Baly, Alberto Barrón-Cedeño, Giovanni Da San Martino, Mitra Mohtarami, Preslav Nakov, and James Glass. 2019. Team QCRI-MIT at SemEval-2019 task 4: Propaganda analysis meets hyperpartisan news detection. In **Proceedings of the 13th International Workshop on Semantic Evaluation**, pages 1041–1046, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Michael Schaffrath. 2003. Mehr als 1:0! Bedeutung des Live-Kommentars bei Fußballübertragungen– eine explorative Fallstudie [more than 1:0! the importance of live commentary on football matches – an exploratory case study]. **Medien und Kommunikationswissenschaft**, 51.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In **Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence**.

C. Shannon. 1948. A mathematical theory of communication. **Bell system technical journal**, 27.

Francesca Stramandinoli, Davide Marocco, and Angelo Cangelosi. 2012. The grounding of higher order concepts in action and language: a cognitive robotics model. **Neural Networks**, 32:165–173.

Yuuya Sugita and Jun Tani. 2005. Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. **Adaptive Behavior**, 13(1):33–52.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In **Advances in Neural Information Processing Systems**, volume 27. Curran Associates, Inc.

Masatoshi Suzuki and Ryo Takahashi. 2021. Japanese bert. https://github.com/cl-tohoku/bert-japanese.

Kumiko Tanaka-Ishii, Koiti Hasida, and Itsuki Noda. 1998. Reactive content selection in the generation of real-time soccer commentary. In **36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2**, pages 1282–1288, Montreal, Quebec, Canada. Association for Computational Linguistics.

Yasufumi Taniguchi, Yukun Feng, Hiroya Takamura, and Manabu Okumura. 2019. Generating Live Soccer-Match Commentary from Play Data. **Proceedings of the AAAI Conference on Artificial Intelligence**, 33(01):7096–7103.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In **Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)**, ICCV '15, pages 4489–4497, USA. IEEE Computer Society.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. In **Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)**, page 125.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In

**Advances in Neural Information Processing Systems**, volume 30, pages 5998–6008. Curran Associates, Inc.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015a. Sequence to sequence – video to text. In **2015 IEEE International Conference on Computer Vision (ICCV)**, pages 4534–4542, Los Alamitos, CA, USA. IEEE Computer Society.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015b. Translating videos to natural language using deep recurrent neural networks. In **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pages 1494–1504, Denver, Colorado. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015a. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, **Advances in Neural Information Processing Systems 28**, pages 2692–2700. Curran Associates, Inc.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. Show and tell: A neural image caption generator. In **Proceedings of the IEEE conference on computer vision and pattern recognition**, pages 3156–3164.

Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. 2018. Reconstruction network for video captioning. In **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, pages 7622–7631, Los Alamitos, CA, USA. IEEE Computer Society.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. 2021. End-to-End Dense Video Captioning With Parallel Decoding. In **Proceedings of the IEEE/CVF International Conference on Computer Vision**, pages 6847–6857.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In **International conference on machine learning**, pages 2048–2057. PMLR.

Yuichi Yamashita and Jun Tani. 2008. Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. **PLoS Comput Biol**, 4(11):e1000220.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2017. Reference-aware language models. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pages 1850–1859, Copenhagen, Denmark. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. **Transactions of the Association for Computational Linguistics**, 2:67–78.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018a. Towards automatic learning of procedures from web instructional videos. In **AAAI Conference on Artificial Intelligence**.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018b. End-to-End Dense Video Captioning With Masked Transformer. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, pages 8739–8748.

仁 松原. 2020. 人工知能と認知科学. **認知科学**, advpub:2020.001.

# 付 録A　発表業績

## A.1　学会誌

- 濵園侑美、小林一郎、麻生英樹、中村友昭、長井隆行、持橋大地,“ヒューマノイドロボットを用いた言語理解による動作生”. 知識と情報、日本知能情報ファジィ学会、32 巻 1 号 p. 632-642、2020.

## A.2　査読付き国際会議

- Yumi Hamazono,Yui Uehara,Hiroshi Noji,Yusuke Miyao,Hiroya Takamura and Ichiro Kobayashi,“Market Comment Generation from Data with Noisy Alignments”, The 13th International Conference on Natural Language Generation, Dec. 2020

- Yumi Hamazono,Tatsuya Ishigaki,Yusuke Miyao,Hiroya Takamura,Ichiro Kobayashi, “Unpredictable Attributes in Market Comment Generation”, The 35th Pacific Asia Conference on Language, Information and Computation, Nov. 2021

- Yumi Hamazono,Tatsuya Ishigaki,Yusuke Miyao,Hiroya Takamura,Ichiro Kobayashi, “Unpredictable Attributes in Market Comment Generation”, The 2022 Conference on Empirical Methods in Natural Language Processing, Dec. 2022

## A.3　国内学会

- 濵園侑美, 上原由衣, 能地宏, 宮尾祐介, 高村大也, 小林一郎，“時系列データの動向説明文生成における参照時刻の不整合解消に向けた取り組み”，言語処理学会第 26 回年次大会，2020.

- 濵園侑美, 上原由衣, 能地宏, 宮尾祐介, 高村大也, 小林一郎, “時系列データの動向説明文生成における参照時刻の不整合解消の精度向上にむけた取り組み”, 言語処理学会第 26 回年次大会, 2020.

# 付 録B　使用したプログラム

## B.1　自然言語からのロボット動作生成

### B.1.1　動作作成プログラム

```
# -*- coding: utf-8 -*-
import sys
import string
import numpy.random as rm
import math

furu = [0, 0, 3.5, 0.5, 0, 0, -3.5, 0.5]
furuu = [0, 3.5, 0, 0.5, 0, -3.5, 0, 0.5]
itameru = [3.5, 0, 0, 0.5, -3.5, 0, 3, 0.5, 0, 0 ,-3, 0.5]
kiru = [0, 0, -5, 0.5, 0, 5, 3.5, 0.5]
mazeru = [0, 3.5, 0, 0.5, 3.5, -3.5, 0, 0.5, -3.5, 0, 0, 0.5]
tsubusu = [0, 0, -3.5, 0.5, 0, 0, 3.5, 0.5]

def degree(bsn, adv_n):
    num = 0
    if bsn == "big":
        while num <= 4.5 + adv_n or num >= 7.5 + adv_n :
            num = rm.normal(6 + adv_n, 1)
        return num

    elif bsn == "small":
        while num <= 0.5 + (adv_n)*0.5 or num >= 2.5 + (adv_n)*0.5:
            num = rm.normal(1 + (adv_n)*0.5, 0.8)
        return num

    elif bsn == "normal":
        num = 3.5
        return num

def speed(fsn, adv_n):
    spd = 0
    if fsn == "fast":
        while spd <= 6.5 + adv_n or spd >= 8 + adv_n:
            spd = rm.normal(6 + adv_n ,1)
        return spd

    elif fsn == "slow":
        while spd <= 0.5 + (adv_n)*0.5 or spd >= 3 + (adv_n)*0.5:
            spd = rm.normal(2 + (adv_n)*0.5, 1)
        return spd

    elif fsn == "normal":
        spd = 5
```

```
44          return spd
45
46  def change_num(adv):
47      if adv>=0 and adv<=2:
48          num = degree("big", adv)
49          spd = speed("fast", adv)
50      elif adv == 3:
51          adv_n = 3-3
52          num = degree("big", adv_n)
53          spd = speed("normal", adv_n)
54      elif adv >= 4 and adv<=5:
55          adv_n = adv - 4
56          num = degree("normal", adv_n)
57          spd = speed("fast", adv_n)
58      elif adv >= 6 and adv <= 8:
59          adv_n = adv - 6
60          num = degree("small", adv_n)
61          spd = speed("normal", adv_n)
62      elif adv >= 9 and adv <= 11:
63          adv_n = adv - 9
64          num = degree("small", adv_n)
65          spd = speed("slow", adv_n)
66      return round(num, 6), round(spd, 6)
67
68  def save_list(list, verb, adv, count):
69      k = 0
70      tanh_list = []
71      while k < len(list):
72          nums = list[k]
73          if k % 4 == 3:
74              newnum =round((math.tanh(nums*0.2)), 6)
75              strnum=str(newnum)
76          else:
77              newnum = round((math.tanh(nums*0.2))*0.5+0.5, 6)
78              strnum=str(newnum)
79          tanh_list.append(strnum)
80          k+=1
81      if len(tanh_list)<=8:
82          act = "\t".join(tanh_list)
83          tanh_act = act + "\t" + act +"\t" + act
84      else:
85          act = "\t".join(tanh_list)
86          tanh_act = act + "\t" + act
87
88      model = "%d_%d_%d.txt" % (verb, adv, count)
89      #print(model)
90      with open(model, "w") as f:
91          f.write(tanh_act)
92      return True
93
94  adv_list = [[0, 2, 3, 4, 6, 8, 10], [1, 3, 4, 5, 7, 10, 11], [2, 3, 4, 7, 8, 9,
        11], \
95      [1, 2, 3, 5, 6, 9, 10], [0, 1, 3, 4, 6, 7, 9], [0, 3, 5, 7, 8, 9, 11]]
96  #verb_list = [[0., 0., -1.*num, spd, 0., 0., num, spd],\
97       [0., num, 0., spd, 0., -1.*num, 0., spd], \
98      [num, 0., 0., spd, -1.*num, 0., 3., spd, 0., 0., -3., spd],\
99       [0., 0., -5., spd, 0., 5., num, spd], \
100     [0., num, 0., spd, num, -1.*num, 0, spd, -1.*num, 0, 0, spd], \
101     [0, 0, -1.*num, spd, 0, 0, num, spd]]
102 j=0
103 for verb_adv in adv_list:
104     for adv in verb_adv:
105         i = 0
```

```
106         while i < 100:
107             num, spd = change_num(adv)
108             verb_list = [[0., 0., num, spd, 0., 0., -1.*num, spd],\
109              [0., num, 0., spd, 0., -1.*num, 0., spd], \
110             [num, 0., 0., spd, -1.*num, 0., 3., spd, 0., 0., -3., spd],\
111              [0., 0., -5., spd, 0., num, 5., spd], \
112             [0., num, 0., spd, num, -1.*num, 0, spd, -1.*num, 0, 0, spd],\
113              [0, 0, -1.*num, spd, 0, 0, num, spd]]
114             #furu = [0., 0., -1.*num, spd, 0., 0., num, spd]
115             verb = verb_list[j]
116             save_list(verb, j, adv, i)
117             i+=1
118     j+=1
```

### B.1.2 ロボット動作プログラム

```
1  #!/opt/grx/bin/hrpsyspy
2  import os
3  import sys
4  import socket
5  import math
6  import time
7  import rtm
8  import waitInput
9  import bodyinfo
10 import java.lang.System
11 import org.omg.CORBA.DoubleHolder
12 import OpenHRP
13 from OpenHRP.RobotHardwareServicePackage import SwitchStatus
14 from OpenHRP.RobotHardwareServicePackage import RobotStateHolder
15 import sample
16 import test
17
18 def init(host='localhost'):
19 #    global vs, vs_svc, cvp, cvp_svc
20 #    global vs_head, vs_head_svc, cvp_head, cvp_head_svc
21
22     sample.init(host)
23
24 def kugirimatome():
25     ugoki = raw_input("What is the name of ugoki?>")
26     ugokilist =[]
27     JAD = []
28     while True:
29         putJAD = input("get JointAnglesDeg? (if fisnish, put 0)> ")
30         if putJAD == 0:
31             break
32         else:
33
34             JAD = sample.getJointAnglesDeg()
35             list1 = []
36             Body = []
37             ArmR = []
38             ArmL = []
39             HandR = []
40             HandL = []
41             newJAD = []#same format as bodyinfo.py
42
```

```
43              for num in JAD:
44                  list1.append(round(num,1))
45
46              Body = list1[0:3]
47              ArmR = list1[3:9]
48              ArmL = list1[9:15]
49              HandR = list1[15:19]
50              HandL = list1[19:23]
51              newJAD = [Body, ArmR, ArmL, HandR, HandL]
52              ttm = int(input("How fast?> "))
53              newugoki = [newJAD, ttm]
54              ugokilist.append(newugoki)
55
56
57      print "%s = %r" %(ugoki,ugokilist)
58
59  def kihon(sa):
60      ugoki = raw_input("What is the name of kihon?>")
61      ugokilist =[]
62      JAD = []
63      while True:
64          putJAD = input("get JointAnglesDeg? (if fisnish, put 0)> ")
65          if putJAD == 0:
66              break
67          else:
68
69              JAD = sa
70              list1 = []
71              Body = []
72              ArmR = []
73              ArmL = []
74              HandR = []
75              HandL = []
76              newJAD = []#same format as bodyinfo.py
77
78              for num in JAD:
79                  list1.append(round(num,1))
80
81              Body = list1[0:3]
82              ArmR = list1[3:9]
83              ArmL = list1[9:15]
84              HandR = list1[15:19]
85              HandL = list1[19:23]
86              newJAD = [Body, ArmR, ArmL, HandR, HandL]
87              ttm = int(input("How fast?> "))
88              newugoki = [newJAD, ttm]
89              ugokilist.append(newugoki)
90
91
92      print "%s = %r" %(ugoki,ugokilist)
93
94  def wake(ugoki):
95      JAD = ugoki
96      Body = []
97      ArmR = []
98      ArmL = []
99      HandR = []
100     HandL = []
101     newJAD = []#same format as bodyinfo.py
102     Body = JAD[0:3]
103     ArmR = JAD[3:9]
104     ArmL = JAD[9:15]
105     HandR = JAD[15:19]
```

```
106     HandL = JAD[19:23]
107     newJAD = [Body, ArmR, ArmL, HandR, HandL]
108     return newJAD
109
110
111
112 def base_use():
113     fans=raw_input("get first JointAnglesDeg?(yes/no):")
114     while fans!="yes":
115         fans=raw_input("get first JointAnglesDeg?(yes/no):")
116     first=sample.getJointAnglesDeg()
117
118     sans=raw_input("get second JointAnglesDeg?(yes/no):")
119     while sans!="yes":
120         sans=raw_input("get second JointAnglesDeg?(yes/no):")
121     second=sample.getJointAnglesDeg()
122
123     sa=[round((y-x),1) for (x, y) in zip(first,second)]
124     return sa
125
126 def mkaction(s):
127     x=test.x
128     y=test.y
129     z=test.z
130     i=0
131     ugokis=[]
132     ugokiss=[]
133     ugoki=sample.getJointAnglesDeg()
134     while i!=len(s):
135         #print s[i][3]
136         xd=[h*s[i][0] for h in x]
137         yd=[j*s[i][1] for j in y]
138         zd=[k*s[i][2] for k in z]
139         ugoki=[round((xs+ys+zs+ugoki),1)for (xs,ys,zs,ugoki) in zip(xd,yd,zd,ugoki)
                ]
140         ugokis=wake(ugoki)
141         #print ugoki
142         ugokis=[ugokis,(s[i][3])]
143         ugokiss=ugokiss+[ugokis]
144         print ugokiss
145         i=i+1
146     return ugokiss
147
148
149 if __name__ == '__main__' or __name__ == 'main':
150   if len(sys.argv) > 1:
151     robotHost = sys.argv[1]
152   else:
153     robotHost = None
154   init(robotHost)
155 #  sample.showJointAnglesDeg()
156 #  kugirimatome()
157
158   sample.goInitial()
159   s=[[0,0,-5,3],[0,0.5,5,3],[0,0,-5,3]] 時系列#で作ったものを入力とするAAM
160   r=mkaction(s)
161   dblHolder = org.omg.CORBA.DoubleHolder()
162   for p in r:
163       sample.setJointAnglesDeg(p[0], p[1], dblHolder)
164
165   sample.goInitial()
```

### B.1.3 ロボット動作プログラム

```
# -*- coding: utf-8 -*-
import sys
import string
import numpy.random as rm
import math

furu = [0, 0, 3.5, 0.5, 0, 0, -3.5, 0.5]
furuu = [0, 3.5, 0, 0.5, 0, -3.5, 0, 0.5]
itameru = [3.5, 0, 0, 0.5, -3.5, 0, 3, 0.5, 0, 0 ,-3, 0.5]
kiru = [0, 0, -5, 0.5, 0, 5, 3.5, 0.5]
mazeru = [0, 3.5, 0, 0.5, 3.5, -3.5, 0, 0.5, -3.5, 0, 0, 0.5]
tsubusu = [0, 0, -3.5, 0.5, 0, 0, 3.5, 0.5]

def degree(bsn, adv_n):
    num = 0
    if bsn == "big":
        while num <= 4.5 + adv_n or num >= 7.5 + adv_n :
            num = rm.normal(6 + adv_n, 1)
        return num

    elif bsn == "small":
        while num <= 0.5 + (adv_n)*0.5 or num >= 2.5 + (adv_n)*0.5:
            num = rm.normal(1 + (adv_n)*0.5, 0.8)
        return num

    elif bsn == "normal":
        num = 3.5
        return num

def speed(fsn, adv_n):
    spd = 0
    if fsn == "fast":
        while spd <= 6.5 + adv_n or spd >= 8 + adv_n:
            spd = rm.normal(6 + adv_n ,1)
        return spd

    elif fsn == "slow":
        while spd <= 0.5 + (adv_n)*0.5 or spd >= 3 + (adv_n)*0.5:
            spd = rm.normal(2 + (adv_n)*0.5, 1)
        return spd

    elif fsn == "normal":
        spd = 5
        return spd

def change_num(adv):
    if adv>=0 and adv<=2:
        num = degree("big", adv)
        spd = speed("fast", adv)
    elif adv == 3:
        adv_n = 3-3
        num = degree("big", adv_n)
        spd = speed("normal", adv_n)
    elif adv >= 4 and adv<=5:
        adv_n = adv - 4
        num = degree("normal", adv_n)
        spd = speed("fast", adv_n)
    elif adv >= 6 and adv <= 8:
        adv_n = adv - 6
        num = degree("small", adv_n)
```

```
61          spd = speed("normal", adv_n)
62      elif adv >= 9 and adv <= 11:
63          adv_n = adv - 9
64          num = degree("small", adv_n)
65          spd = speed("slow", adv_n)
66      return round(num, 6), round(spd, 6)
67
68  def save_list(list, verb, adv, count):
69      k = 0
70      tanh_list = []
71      while k < len(list):
72          nums = list[k]
73          if k % 4 == 3:
74              newnum =round((math.tanh(nums*0.2)), 6)
75              strnum=str(newnum)
76          else:
77              newnum = round((math.tanh(nums*0.2))*0.5+0.5, 6)
78              strnum=str(newnum)
79          tanh_list.append(strnum)
80          k+=1
81      if len(tanh_list)<=8:
82          act = "\t".join(tanh_list)
83          tanh_act = act + "\t" + act +"\t" + act
84      else:
85          act = "\t".join(tanh_list)
86          tanh_act = act + "\t" + act
87
88      model = "%d_%d_%d.txt" % (verb, adv, count)
89      #print(model)
90      with open(model, "w") as f:
91          f.write(tanh_act)
92      return True
93
94  adv_list = [[0, 2, 3, 4, 6, 8, 10], [1, 3, 4, 5, 7, 10, 11], [2, 3, 4, 7, 8, 9,
        11], \
95      [1, 2, 3, 5, 6, 9, 10], [0, 1, 3, 4, 6, 7, 9], [0, 3, 5, 7, 8, 9, 11]]
96  #verb_list = [[0., 0., -1.*num, spd, 0., 0., num, spd],\
97       [0., num, 0., spd, 0., -1.*num, 0., spd], \
98      [num, 0., 0., spd, -1.*num, 0., 3., spd, 0., 0., -3., spd],\
99       [0., 0., -5., spd, 0., 5., num, spd], \
100     [0., num, 0., spd, num, -1.*num, 0, spd, -1.*num, 0, 0, spd], \
101     [0, 0, -1.*num, spd, 0, 0, num, spd]]
102 j=0
103 for verb_adv in adv_list:
104     for adv in verb_adv:
105         i = 0
106         while i < 100:
107             num, spd = change_num(adv)
108             verb_list = [[0., 0., num, spd, 0., 0., -1.*num, spd],\
109              [0., num, 0., spd, 0., -1.*num, 0., spd], \
110             [num, 0., 0., spd, -1.*num, 0., 3., spd, 0., 0., -3., spd],\
111              [0., 0., -5., spd, 0., num, 5., spd], \
112             [0., num, 0., spd, num, -1.*num, 0, spd, -1.*num, 0, 0, spd],\
113              [0, 0, -1.*num, spd, 0, 0, num, spd]]
114             #furu = [0., 0., -1.*num, spd, 0., 0., num, spd]
115             verb = verb_list[j]
116             save_list(verb, j, adv, i)
117             i+=1
118     j+=1
```

### B.1.4 cookpad コーパス MySQL から word2vec が使える形への変換

```
\begin{small}
\begin{verbatim}
# -*- coding:utf-8 -*-
import mysql.connector
import re
import MeCab
import unicodedata

tagger = MeCab.Tagger('mecabrc -d /usr/local/lib/mecab/dic/mecab-ipadic-neologd')
line_list = []


def make_sentence(line_list):
    if not line_list[len(line_list)-1][1] == 読点"" and\
            line_list[len(line_list)-1][0] == 記号"" or\
            line_list[len(line_list)-1][1] == 終助詞"" or \
            line_list[len(line_list)-1][2] == 基本形"" or \命令
                    "" in line_list[len(line_list)-1][2] or \
                    line_list[len(line_list)-1][0] == 名詞"":
        sentence = ""
        for l in line_list:
            sentence += (l[3]) + " "
        print(sentence)
        return True

def wakachi(line):
    global line_list
    node = tagger.parseToNode(line)
    node = node.next
    while node.next:
        ftr = node.feature.split(",")
        #print(ftr[0], ftr[1], ftr[5], ftr[6])
        if ftr[6] == "*":
            ftr[6] = node.surface
        if ftr[0] == 名詞'':
            #print(node.surface)
            ftr[6] = node.surface
        line_list.append([ftr[0], ftr[1], ftr[5], ftr[6]])
        node = node.next
    #print(line_list)
    if make_sentence(line_list) is True:
            line_list = []
    #print(line_list)

def rm_return(line):
    list = re.sub('[\t\r\f\v]', '', line.strip())
    return list



def normalise(s):
    s = unicodedata.normalize('NFKC', s)正規化する文字化け
    #()
    s = re.sub('[\UTF{02D7}\UTF{058A‐}\UTF{2011} \
      \UTF{2012}\UTF{2013}\UTF{2043}\UTF{207B}\UTF{208B−}+]', '-', s)
    s = re.sub('[\UTF{FE63}\UTF{FF0D}^^ef^^bd^^ーーーーー−b0]+', ー'', s)
    s = re.sub('[~\UTF{223C}\UTF{223E〜}\UTF{3030}\UTF{FF5E}]', '', s)
    s = re.sub('[\UTF{2460}-\UTF{2468}]+', '', s)
    s = re.sub℃('', 度'', s)
    s = re.sub, ('', 、'', s)
```

```
61      s = re.sub. (’’, 。’’, s)
62      s = re.sub(’[\UTF{266C}\UTF♪{2764}\UTF{261D}\UTF{2661}\UTF{266B}\UTF{263A◆
            }\UTF{2765}\CID{115}]’, ’\n’, s)
63      s = re.sub(’\UTF{273F}’, ’’, s)
64      s = s.strip(’\x00’)
65      s = s.replace° (’C’, 度’’)
66      return s
67
68
69  def put_return(s):
70      s = re.sub。(’’, 。’\n’, s)
71      list = s.split(’\n’)
72      for l in list:
73          l.strip()
74          if not l == ’’ and not l == ’ ’:
75              #print(l)
76              wakachi(l)
77
78
79  def get_word_mysql(db_name, pw):
80      con = mysql.connector.connect(host=’localhost’, db=db_name, user=’root’, \
81      passwd=pw, buffered=True)
82      cur = con.cursor()
83      select = "select memo from steps where memo like \一口大
84       ’%%’"  # where memo like 切’%%’"
85      cur.execute(select)
86      res = cur.fetchall()
87      for row in res:
88          if not row[0] == None and not row[0] == ’’: # and "\UTF{2764}" in row[0]:
89              #print(row)
90              put_return(rm_return(normalise(row[0])))
91      cur.close
92      con.close
93
94  def get_recipe_id(db_name, pw):
95      con = mysql.connector.connect(host=’localhost’, db=db_name, user=’root’, \
96      passwd=pw, buffered=True)
97      cur = con.cursor()
98      select = "select recipe_id from steps"  # where memo like 切’%%’"
99      cur.execute(select)
100     res = cur.fetchall()
101     min = ""
102     max = ""
103     for row in res:
104         if min == "":
105             min = row[0]
106         if max == "":
107             max = row[0]
108         else:
109             max = row[0]
110     print(min, max)
111     cur.close
112     con.close
113
114
115 if __name__ == "__main__":
116     get_word_mysql(’cookpad_data’, pw) #はのパスワードを入力pwmysql
117     #get_recipe_id(’cookpad_data’, pw)
```

### B.1.5 ネットワークの設計プログラム

Listing B.1: Net2

```
# -*- coding: utf-8 -*-
import chainer
import chainer.functions as F
import chainer.links as L

class act_prediction(chainer.Chain):
    def __init__(self, verb_in, adv_in, unit, act_out):
        super(act_prediction, self).__init__(
            verb_l1 = L.Linear(verb_in, unit),
            verb_l2 = L.Linear(unit, unit),
            adv_l1 = L.Linear(adv_in, unit),
            adv_l2 = L.Linear(unit, unit),
            l3 = L.Linear(unit, act_out),
        )
    def softplus(self, x):
        return F.log(F.exp(x) + 1)

    def __call__(self, verb, adv):
        verb_h1 = self.softplus(self.verb_l1(verb))
        verb_h2 = self.softplus(self.verb_l2(verb_h1))

        adv_h1 = self.softplus(self.adv_l1(adv))
        adv_h2 = self.softplus(self.adv_l2(adv_h1))

        output = F.sigmoid(self.l3(verb_h2 + adv_h2))

        return output
```

Listing B.2: Net3

```
# -*- coding: utf-8 -*-
import chainer
import chainer.functions as F
import chainer.links as L

class act_prediction(chainer.Chain):
    def __init__(self, verb_in, adv_in, unit, act_out):
        super(act_prediction, self).__init__(
            verb_l1 = L.Linear(verb_in, unit),

            adv_l1 = L.Linear(adv_in, unit),

            l2 = L.Linear(unit, unit),
            l3 = L.Linear(unit, act_out),
        )
    def softplus(self, x):
        return F.log(F.exp(x) + 1)

    def __call__(self, verb, adv):
        verb_h1 = self.softplus(self.verb_l1(verb))

        adv_h1 = self.softplus(self.adv_l1(adv))

        h2 = self.softplus(self.l2(verb_h1 + adv_h1))
        output = F.sigmoid(self.l3(h2))

        return output
```

Listing B.3: Net4

```
# -*- coding: utf-8 -*-
import chainer
import chainer.functions as F
import chainer.links as L

class act_prediction(chainer.Chain):
    def __init__(self, verb_in, adv_in, unit, act_out):
        super(act_prediction, self).__init__(
            verb_l1 = L.Linear(verb_in, unit),
            adv_l1 = L.Linear(adv_in, unit),
            adv_verb_l2 = L.Linear(unit, unit),
            l2 = L.Linear(unit, unit),
            l3 = L.Linear(unit, act_out),
        )
    def softplus(self, x):
        return F.log(F.exp(x) + 1)

    def __call__(self, verb, adv):

        verb_h1 = self.softplus(self.verb_l1(verb))

        adv_h1 = self.softplus(self.adv_l1(adv))
        adv_verb_h2 = self.softplus(self.adv_verb_l2(verb_h1 + adv_h1))

        h2 = self.softplus(self.l2(verb_h1 + adv_verb_h2))
        output = F.sigmoid(self.l3(h2))

        return output
```

### B.1.6 訓練プログラム

```
# -*- coding: utf-8 -*-

from __future__ import print_function
import argparse

import numpy as np
import six
#import net3 as net
#import net2 as net
import net1 as net 使うネットワークに合わせて#を考えるimport
import read_file
import time
import chainer
import chainer.functions as F
from chainer import optimizers
from chainer import serializers

import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
import os

```

```
parser = argparse.ArgumentParser(description='NN for act prediction')
parser.add_argument('--gpu', '-g', default=-1, type=int,
                    help='GPU ID (negative value indicates CPU)')
#数epoch
parser.add_argument('--epoch', '-e', default=20, type=int,
                    help='number of epochs to learn')中間層のノード数
#
parser.add_argument('--unit', '-u', default=14, type=int,
                    help='number of units')動作入力層のノード数
#
parser.add_argument('--verb_input', '-vi', default=6, type=int,
                    help='number of input')副詞入力層のノード数
#
parser.add_argument('--adv_input', '-ai', default=12, type=int,
                    help='number of input')バッチ数
#
parser.add_argument('--batchsize', '-b', type=int, default=100,
                    help='learning minibatch size')

parser.add_argument('--dataset_dir', "-d", default = 'dataset1601',
                    help='Path to dataset')

parser.add_argument('--save_dir', "-sd", default='model',
                    help='Path to model name')ネットワークの種類

#
#parser.add_argument('--network_name', "-n", default='net',
#                     help='Path to net name')

args = parser.parse_args()
if not os.path.exists(args.save_dir):
    os.mkdir(args.save_dir)
batchsize = args.batchsize  # default:100
n_epoch = args.epoch        # default:20
n_units = args.unit         # default:14
verb_in = args.verb_input   # default:6
adv_in = args.adv_input     # default:12
dataset_dir = args.dataset_dir
save_dir = args.save_dir
#network = args.network_name


print('GPU: {}'.format(args.gpu))
print('# unit: {}'.format(args.unit))
print('# Minibatch-size: {}'.format(args.batchsize))
print('# epoch: {}'.format(args.epoch))
print('')データ準備


#
print('load dataset...')
all_verb, all_adv, all_act = read_file.read_file(dataset_dir)
print('Done')
#print(len(all_verb))
N = 3800
train_verb = all_verb[:N]
train_adv = all_adv[:N]
train_act = all_act[:N]

test_verb = all_verb[N:]
test_adv = all_adv[N:]
test_act = all_act[N:]

```

```
N_test = len(test_act)モデルセット

#
mlp = net.act_prediction(verb_in, adv_in, n_units, 24)

# GPU を使う場合
if args.gpu >= 0:
    chainer.cuda.get_device(args.gpu).use()
    mlp.to_gpu()「
#はxp, ならCPUnp, ならとする」というお得意の呪文GPUcupychainer
xp = np if args.gpu < 0 else chainer.cuda.cupy

# Setup optimizer
optimizer = optimizers.Adam()
#optimizer = optimizers.SGD(lr=0.001) # 最適化関数はを使用Adam
optimizer.setup(mlp) # optimizer に model をセット
# Learning loop
train_loss = []
test_loss = []
for epoch in six.moves.range(1, n_epoch + 1):
    #print('epoch', epoch)
    ### training ###
    perm = np.random.permutation(N)  # 個の数字をランダムな順に並べるN(shuffle)
    sum_loss = 0
    start = time.time() # 計算時間計測開始
    for i in six.moves.range(0, N, batchsize):
        verb = chainer.Variable(xp.asarray(train_verb[perm[i:i + batchsize]]))
        adv = chainer.Variable(xp.asarray(train_adv[perm[i:i + batchsize]]))
        act = chainer.Variable(xp.asarray(train_act[perm[i:i + batchsize]]))

        optimizer.zero_grads()
        #output, loss = mlp(verb, adv, act)
        loss = F.mean_squared_error(mlp(verb, adv), act)
        loss.backward()
        optimizer.update()
        sum_loss += float(loss.data)*len(act.data)
        end = time.time()

    elapsed_time = end - start
    throughput = N / elapsed_time
    train_loss.append(sum_loss/N)
    #print('train mean loss = {}, throughput = {},\
     images/sec'.format(sum_loss/N, throughput))


    #test
    sum_loss = 0
    for i in six.moves.range(0, N_test, batchsize):
        verb = chainer.Variable(xp.asarray(train_verb[perm[i:i + batchsize]]),\
         volatile='on') # したくないときはdorpouton
        adv = chainer.Variable(xp.asarray(train_adv[perm[i:i + batchsize]]),\
         volatile='on')
        act = chainer.Variable(xp.asarray(train_act[perm[i:i + batchsize]]),\
         volatile='on')
        loss = F.mean_squared_error(mlp(verb, adv), act)
        # とだけ計算forwardloss,の必要なしbackward

        sum_loss += float(loss.data) * len(act.data)
        # sum_accuracy += float(model.accuracy.data) * len(t.data)
        # print('test  mean loss={}, \
        accuracy={}'.format(sum_loss / N_test, sum_accuracy / N_test))
    test_loss.append(sum_loss/N_test)
    #print('test mean loss={},\
```

```
    throughput={} images/sec'.format(sum_loss / N_test, throughput))

    if epoch % 1000 == 0:
        model_path = "%s/act_In_verb_adv_%d.model" % (save_dir, epoch)
        #with open(model_path, "w") as f:
        serializers.save_npz(model_path, mlp)

        state_path = "%s/act_In_verb_adv_%d.state" % (save_dir, epoch)
        #with open(state_path, "w") as f:
        serializers.save_npz(state_path, optimizer)


# Save the model and the optimizer
print('save the model')
model_path = "%s/act_In_verb_adv_%d.model" % (save_dir, epoch)
#with open(model_path, "w") as f:
serializers.save_npz(model_path, mlp)

print('save the optimizer')
state_path = "%s/act_In_verb_adv_%d.state" % (save_dir, epoch)
#with open(state_path, "w") as f:
serializers.save_npz(state_path, optimizer)


plt.figure(figsize=(8,12))
plt.subplot(2, 1, 1)
plt.title("Loss of Train Data Recognition.")
plt.ylim(ymax=round(train_loss[0],1))
plt.plot(range(len(train_loss)), train_loss)
plt.text(len(train_loss)-1, train_loss[-1],\
round(train_loss[-1], 6), ha = 'right', va = 'bottom')

plt.subplot(2, 1, 2)
plt.title("Loss of Test Data Recognition.")
#plt.yscale("log")
plt.ylim(ymax=0.1)
plt.plot(range(len(test_loss)), test_loss)
plt.text(len(test_loss)-1, test_loss[-1], \
round(test_loss[-1], 6), ha = 'right', va = 'bottom')
plt.tight_layout()
plt.savefig("%s/loss.png" %(save_dir))
```

### B.1.7 3D 描写プログラム

```
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import matplotlib.pyplot as plt
# を取得Axes3DSubplot

def make_fig(x, y, z, name):
    fig = plt.figure()
    ax = fig.gca(projection='3d')
    Xz, Yz = np.meshgrid(np.arange(0, 8, 7), np.arange(-7, 8, 7))
    Zz = 0
    ax.plot_surface(Xz, Zz, Yz, rstride=1, cstride=1, alpha = 0.1)
    ax.plot_surface(Zz, Xz, Yz, rstride=1, cstride=1, alpha = 0.1)
    line, = ax.plot(x, y, z, zdir='z')
    ax.view_init(15.0, 30.0)
```

```
    plt.axis('scaled')
    fig_name = name + ".png"
    fig.savefig(fig_name)
```

### B.1.8 評価プログラム

```
# -*- coding: utf-8 -*-

from __future__ import print_function
import argparse

import numpy as np
import six
import os
import time
import chainer
import chainer.functions as F
from chainer import optimizers
from chainer import serializers
import matplotlib.pyplot as plt
import math
import mkfig #3に描写D
import mkdefact_0112 as mkdefact デフォルトの動作を作成#
import read_file
#import net1 as net ネットワークに合わせて選ぶ#
#import net2 as net
#import net3 as net


parser = argparse.ArgumentParser(description='NN for act prediction')
parser.add_argument('--gpu', '-g', default=-1, type=int,
                    help='GPU ID (negative value indicates CPU)')
#数epoch
#parser.add_argument('--epoch', '-e', default=20, type=int,
#                    help='number of epochs to learn')中間層のノード数

#
parser.add_argument('--unit', '-u', default=14, type=int,
                    help='number of units')動作入力層のノード数
#
parser.add_argument('--verb_input', '-vi', default=6, type=int,
                    help='number of input')副詞入力層のノード数
#
parser.add_argument('--adv_input', '-ai', default=12, type=int,
                    help='number of input')学習モデル

#
parser.add_argument('--model', "-m", default = 'mlp',
                    help='Path to trained all movement model')ネットワークの種類

#
#parser.add_argument('--network_name', "-n", default='net',
#                    help='Path to net name')

args = parser.parse_args()
if not os.path.exists(args.model):
    os.mkdir(args.model)
#batchsize = args.batchsize  # default:100
```

```
#n_epoch = args.epoch        # default:20
n_units = args.unit          # default:1000
verb_in = args.verb_input    # default:6
adv_in = args.adv_input      # default:12
model = args.model + ".model"

print('GPU: {}'.format(args.gpu))
print('# unit: {}'.format(args.unit))
#print('# Minibatch-size: {}'.format(args.batchsize))
#print('# epoch: {}'.format(args.epoch))
print('')

if "share" in model:
    import net_share as net
elif "hid" in model:
    import net_hid2 as net
elif "sig" in model:
    import net_sig as net
elif "sft" in model:
    import net_softplus as net

mlp = net.act_prediction(verb_in, adv_in, n_units, 24)
serializers.load_npz(model, mlp)

loss_path = "%s/loss_result.txt" % (args.model)
fl = open(loss_path, "w")


# GPU を使う場合
if args.gpu >= 0:
    chainer.cuda.get_device(args.gpu).use()
    mlp.to_gpu()「

#はxp, ならCPUnp, ならとする」というお得意の呪文GPUcupychainer
xp = np if args.gpu < 0 else chainer.cuda.cupy

verb_ja = ふる["", ふるう"", 炒める"", 切る"", 混ぜる"", つぶす""]
adv_ja = さっと["", ざっと"", ザクザク"", たっぷり"", 手早い"", \一気に
"", 少し"", 細かい"", ちょっと"", しっかり"", きちんと"", じっくり""]
defomove = mkdefact.mkdef()
#print(defomove)
sum_traind_loss = 0
sum_non_loss = 0
sum_loss = 0
adv_list = [[0, 2, 3, 4, 6, 8, 10], [1, 3, 4, 5, 7, 10, 11], [2, 3, 4, 7, 8, 9,
    11],\
 [1, 2, 3, 5, 6, 9, 10], [0, 1, 3, 4, 6, 7, 9], [0, 3, 5, 7, 8, 9, 11]]
str_list, vec_list = read_file.read_word_file(word_file)
  #word_file=で作ったベクトルを保存したファイルword2vec

for i in range(len(verb_ja)):
    #np_verb = np.array([0.] * verb_in, dtype=np.float32)
    #np_verb[i] = 1.
    result_path = "%s/%d%s_result.txt" % (args.model, i, verb_ja[i])
    f = open(result_path, "w")
    v_sum_loss = 0
    for j in range(len(adv_ja)):
        #np_adv = np.array([0.] * adv_in, dtype=np.float32)
        #np_adv[j] = 1.
        verb_vec = read_file.str_to_vec(str_list, vec_list, verb_ja[i])
        adv_vec = read_file.str_to_vec(str_list, vec_list, adv_ja[j])
        verb = chainer.Variable(xp.asarray([verb_vec]), volatile='on')
        adv = chainer.Variable(xp.asarray([adv_vec]), volatile='on')
```

```
115         out = mlp(verb, adv)
116         l = (list(out.data))
117         for k in l:
118             num = list(k)
119             t = 0
120             sum = 0
121             while t < len(num):
122                 if t % 4 == 3:
123                     num[t]  =round(0.25*math.log((1+(float(num[t])))/ \
124                     (1-(float(num[t])))),2)
125                 else:
126                     num[t] = round(2.5*math.log((1+((float(num[t])-0.5)*2))\
127                                                 /(1-((float(num[t])-0.5)*2))), 2)
128                 if i == 2 or i == 4:
129                     loss = defomove[j][i][t%12]-num[t]
130                 else:
131                     loss = defomove[j][i][t%8]-num[t]
132                 loss_mean = loss*loss
133                 sum += loss_mean
134                 t += 1
135             #print(i, j)
136             rms = round(math.sqrt(sum/t), 4)
137             if j in adv_list[i]:
138                 sum_traind_loss += rms
139             else:
140                 sum_non_loss += rms
141             sum_loss += rms
142             v_sum_loss += rms
143             #print(verb_ja[i], adv_ja[j], rms)
144             #print([num[r:r+4] for r in range(0, len(num), 4)])
145 
146             num_list = [num[r:r+4] for r in range(0, len(num), 4)]
147 
148             x = [0.]
149             y = [0.]
150             z = [0.]
151             countss=0
152             if i == 2 or i == 4:
153                 while countss < 3:
154                     x.append(x[countss] + num_list[countss][0])
155                     y.append(y[countss] + num_list[countss][1])
156                     z.append(z[countss] + num_list[countss][2])
157                     countss+=1
158             else:
159                 while countss < 2:
160                     x.append(x[countss] + num_list[countss][0])
161                     y.append(y[countss] + num_list[countss][1])
162                     z.append(z[countss] + num_list[countss][2])
163                     countss+=1
164 
165         mkfig.make_fig(x, y, z, args.model+"/"+str(i)+"_"+str(j)+verb_ja[i]+adv_ja[
                j])
166         result_path = "%s/result.txt" % (args.model)
167         #with open(result_path, "a") as f:
168         f.write(verb_ja[i] + "\t"+ adv_ja[j] + "\n")
169         f.write(str(rms)+"\n")
170         f.write(str(num_list) + "\n")
171         fl.write(str(i)+"\t"+str(j)+"\t"+str(rms)+"\n")
172     f.close()
173     fl.write(str(i)+"\t"+str(round(v_sum_loss/12, 4))+ "\n")
174 fl.write("\n" + "all_loss\t"+ str(round(sum_loss/12/6, 4)) + "\n")
175 fl.write("traind_loss\t"+ str(round(sum_traind_loss/7/6, 4))  + "\n")
176 fl.write("nontraind_loss\t"+ str(round(sum_non_loss/5/6, 4))  + "\n")
```

```
177 fl.close()
```

## B.2 一般ドメイン実況生成

### B.2.1 実況コメント録音 UI

Listing B.4: Recording

```
1  <script src="https://sdk.amazonaws.com/js/aws-sdk-2.142.0.min.js"></script><script
       src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script
       >
2  <link href="https://s3.amazonaws.com/mturk-public/bs30/css/bootstrap.min.css" rel="
       stylesheet" />
3  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="
       stylesheet" />
4  <style type="text/css">fieldset {
5      padding: 10px;
6      background: #fbfbfb;
7      border-radius: 5px;
8      margin-bottom: 5px;
9    }
10   .fileUploadFieldset .record_button {
11     display: none;
12   }
13   .d-flex {
14     display: flex;
15   }
16   .ytplayer.nocontrol {
17     pointer-events: none;
18   }
19   .recordway {
20     display: none;
21   }
22 </style>
23 <section class="container" id="Other" style="margin-bottom:15px; padding: 10px 10px
       ; font-family: Verdana, Geneva, sans-serif; color:#333333; font-size:0.9em;">
24 <div class="row col-xs-12 col-md-12"><!-- Instructions -->
25 <fieldset><label>Read the Instructions</label>
26 <div class="panel panel-primary">
27 <div class="panel-heading"><strong>Instructions</strong></div>
28
29 <div class="panel-body">
30 <p><b>Watch the videos, <span class="recordway">and </span>record live commentaries
       <span class="uploadway">, and upload recorded files</span></b></p>
31 <!-- We are adding a template variable below which we will replace with an actual
       link in our code later -->
32
33 <ul>
34   <li>This survey gathers live commentaries of videos.<br />
35   <span class="uploadway">You need some recording tools.</span><br />
```

```
36    You will be shown 2 videos.</li>
37    <br />
38    <li>This HIT allows for repeat participation if the videos are different.</
          li>
39    <li>Before starting, please make sure that following <strong>Video1</strong>
           and <strong>Video2</strong> videos appear as shown in <strong>Example</
          strong>. We have checked the videos before creating this HIT, but
          depending on the timing or your situation, the video may not play. If you
          cannot watch the video, please do not work on this HIT (You will not be able
          to submit it).</li>
40    <li>Please record a live commentary <b>twice</b> for each of the video.
41    <ul>
42      <li>Record a live commentary while watching a video.</li>
43      <li>Record a live commentary again while watching the same video.</li>
44      <li class="uploadway">Upload two recording files.</li>
45      <li class="recordway">The video will start playing as you click the record
            button.</li>
46      <!--
47                  <li>Please make sure the time length of each recorded commentary is
                         within its video's time.</li>
48                  -->
49      <li class="recordway">You will not be able to stop recording before the video
            finishes. The recording will automatically shut off 60 seconds after the
            video ends, if you don&#39;t stop it sooner.</li>
50      <!--
51      <li>You cannot refer to any word in the video.</li>
52                  --><!--
53      <li>Use only English in your live commentaries.</li>
54      <li><b>Keep talking and recording the commentary until the video is over.</b></
            li>
55      <li>Please ignore any written text you see in the video, and describe the video
             in your own words.</li>
56  -->
57      <li>The audio of the video is muted.</li>
58      <li>When recording starts, <strong>(recorded XX:XX/XX:XX)</strong> will be
            displayed next to <strong>First commentary:</strong> or <strong>Second
            commentary:</strong>. If not, your environment is not compatible with this
            HIT, and you should stop working on it.</li>
59      <li><b>Important!</b> (Check them all)
60      <div class="checkbox"><label><input name="Check_important1" required="" type="
            checkbox" value="check_ok" />Keep talking and recording the commentary
            until the video is over.</label></div>
61
62      <div class="checkbox"><label><input name="Check_important2" required="" type="
            checkbox" value="check_ok" />Please ignore any written text you see in the
            video, and describe the video in your own words.</label></div>
63
64      <div class="checkbox"><label><input name="Check_important3" required="" type="
            checkbox" value="check_ok" />Use only English in your live commentaries.</
            label></div>
65
66      <div class="checkbox"><label><input name="Check_important4" required="" type="
            checkbox" value="check_ok" />Please do not start with the phrases like <em
            >&quot;OK, this is the second recoding&quot;</em>, just describe what is on
             the video.</label></div>
67
68      <div class="checkbox"><label><input name="Check_important5" required="" type="
            checkbox" value="check_ok" />If an ad is played while recording, say <em>&
            quot;ad starts&quot;</em> when the ad starts, and <em>&quot;ad ends&quot;</
            em> when it ends.</label></div>
69
70      <div class="checkbox"><label><input name="Check_important6" required="" type="
            checkbox" value="check_ok" /></label>Please check your recordings before
```

```
        submission. The auto-recognition function may reject it when your voice is
        too quiet, there is too much noise, or the total time of the commentary
        audio part only is less than 60% of the video duration.</div>
    </li>
  </ul>
  </li>
  <li>Also answer the questionnaire about the videos.
  <ul>
    <li>How difficult was it to record a live commentary? and why?</li>
    <li>How many scene cuts are there in this video?</li>
    <!--
<li>How fast is each scene cut?</li>
    <li>How many events are there in each scene cut?</li>
-->
  </ul>
  </li>
  <li>Finally answer the questionnaire about your background.
  <ul>
    <li>The country you are living in</li>
    <li>The language(s) you can speak fluently</li>
    <li>Your familiarity with: Sports, Music,...etc</li>
  </ul>
  </li>
</ul>

<section>
<div class="checkbox"><label><input name="Check" required="" type="checkbox" value
    ="check_ok" />Have you checked the instructions?</label></div>
</section>
</div>
</div>

<div class="panel panel-primary">
<div class="panel-heading"><strong>Example</strong></div>

<div class="panel-body d-flex">
<div class="ytplayer" data-id="H6hb3SIqdL4"> </div>

<div>
<fieldset class="fileUploadFieldset"><label>First commentary (sample audio):</label
    ><br />
<input class="fileUpload" disabled="disabled" single="" type="file" /><button class
    ="record_button" disabled="disabled"><i class="material-icons">mic</i></button>

<audio class="playback" controls="" data-loc="sample/sample_c1.mp3" disabled="
    disabled"> </audio>
</fieldset>

<fieldset class="fileUploadFieldset"><label>Second commentary (sample audio):</
    label><br />
<input class="fileUpload" disabled="disabled" single="" type="file" /><button class
    ="record_button" disabled="disabled"><i class="material-icons">mic</i></button>

<audio class="playback" controls="" data-loc="sample/sample_c2.mp3" disabled="
    disabled"> </audio>
</fieldset>
</div>
</div>
</div>
</fieldset>
</div>
<!-- End Instructions --><!-- Content Body --><input id="audiomethod" name="method"
     type="hidden" value="uploaded" />
```

```
<fieldset><label>Video</label>

<div class="panel panel-primary">
<div class="panel-heading"><strong>Video 1</strong></div>

<div class="panel-body d-flex">
<div class="ytplayer" data-id="${VideoId1}" data-set="${Datatype1}"> </div>
<!-- test goat --><!-- <div class="ytplayer" data-id="jRxSRyrTANY" data-set="${
    Datatype1}"> </div> --><input class="videoDuration" name="v1d" type="
    hidden" />
<div>
<fieldset class="fileUploadFieldset"><label>First commentary<span class="timer">:</
    span></label><br />
<input class="audioinput fileUpload" single="" type="file" /><button class="
    audioinput record_button" disabled="disabled"><i class="material-icons">mic</i
    ></button>

<audio class="playback" controls="" disabled="disabled"> </audio>
<input class="duration" data-cid="1" name="v1c1d" type="hidden" /><input class="
    filekey" data-cid="1" name="v1c1" type="hidden" />
<div class="status"> </div>
Did any ads play while the video was playing?<br />
<input checked="checked" name="v1c1cm" required="" type="radio" value="0" /> No &
    nbsp;<input name="v1c1cm" type="radio" value="1" /> Yes</fieldset>

<fieldset class="fileUploadFieldset"><label>Second commentary<span class="timer
    ">:</span></label><br />
<input class="audioinput fileUpload" single="" type="file" /><button class="
    audioinput record_button" disabled="disabled"><i class="material-icons">mic</i
    ></button>

<audio class="playback" controls="" disabled="disabled"> </audio>
<input class="duration" data-cid="2" name="v1c2d" type="hidden" /><input class="
    filekey" data-cid="2" name="v1c2" type="hidden" />
<div class="status"> </div>
Did any ads play while the video was playing?<br />
<input checked="checked" name="v1c2cm" required="" type="radio" value="0" /> No &
    nbsp;<input name="v1c2cm" type="radio" value="1" /> Yes</fieldset>
</div>
</div>

<div class="panel-body">
<p><b>Questionnaire about Video 1</b></p>

<ul>
  <li☆>How difficult was it to record a live commentary with this video? (1:Easy &
      larr; &rarr; 5:Difficult)<br />
  <input name="v1_difficulty" required="" type="radio" value="1" /> 1  <input
      name="v1_difficulty" type="radio" value="2" /> 2  <input name="
      v1_difficulty" type="radio" value="3" /> 3  <input name="v1_difficulty"
      type="radio" value="4" /> 4  <input name="v1_difficulty" type="radio"
      value="5" /> 5 </li>
  <li>Why was it difficult?
  <ul>
    <li>Speed (1:Too slow&larr; &rarr;5:Too fast; Check <b>3</b> if you answered ☆
        as 1:Easy)<br />
    <input name="v1_speed" required="" type="radio" value="1" />1  <input name
        ="v1_speed" type="radio" value="2" /> 2  <input name="v1_speed" type="
        radio" value="3" /> 3  <input name="v1_speed" type="radio" value="4"
        /> 4  <input name="v1_speed" type="radio" value="5" /> 5 </li>
    <li>Event changing in each scene cuts (1:Too few to continue describing&larr; &
        rarr;5:Too many to describe in time; Check <b>3</b> if you answered ☆
        as 1:Easy)<br />
```

```
163     <input name="v1_event" required="" type="radio" value="1" />1  <input name
            ="v1_event" type="radio" value="2" /> 2  <input name="v1_event" type="
            radio" value="3" /> 3  <input name="v1_event" type="radio" value="4"
            /> 4  <input name="v1_event" type="radio" value="5" /> 5 </li>
164     <li>Video content (situation, object, event,...etc)<br />
165     <input name="v1_content" required="" type="radio" value="unknown" /> 
            Unknown  <input name="v1_content" type="radio" value="known" />&
            nbsp;Known  <input name="v1_content" required="" type="radio"
            value="familiar" /> Familiar  </li>
166     <li>
167     <p>Other<br\> <input class="form-control" id="v1_other" name="v1_other"
            placeholder="e.g. 'The screen shakes so much that it is difficult to grasp
            the situation'" type="text" /> </br\></p>
168     </li>
169   </ul>
170   <!--
171   <li>How fast is each scene cut?<br />
172   (1: Scene cuts are too long, 3: Best speed for recording the commentary , 5:
          Scene cuts are too fast)<br />
173   <input name="v1q3" required="" type="radio" value="1" /> 1  <input name="
          v1q3" type="radio" value="2" /> 2  <input name="v1q3" type="radio" value
          ="3" /> 3  <input name="v1q3" type="radio" value="4" /> 4  <input
          name="v1q3" type="radio" value="5" /> 5 </li>
174   <li>How many events are there in each scene cut?<br />
175   (1: The events do not change* at all within one scene cut, 3: Best amount for
          recording the commentary , 5: Too many events within one scene cut)<br />
176   * Event changing is such as a change in the behavior of a person or a change in
          an object in the video.<br />
177   <input name="v1q4" required="" type="radio" value="1" /> 1  <input name="
          v1q4" type="radio" value="2" /> 2  <input name="v1q4" type="radio" value
          ="3" /> 3  <input name="v1q4" type="radio" value="4" /> 4  <input
          name="v1q4" type="radio" value="5" /> 5 </li>
178 --></li>
179   <li>How many scene cuts are there in this video?<br />
180   <input name="v1_scenecuts " required="" type="radio" value="1" /> 1  <input
          name="v1_scenecuts " type="radio" value="2-4" /> 2-4  <input name="
          v1_scenecuts " type="radio" value="5-8" /> 5-8  <input name="
          v1_scenecuts " type="radio" value="9-14" /> 9-14  <input name="
          v1_scenecuts " type="radio" value="over15" /> over 15- </li>
181 </ul>
182 </div>
183 </div>
184
185 <div class="panel panel-primary">
186 <div class="panel-heading"><strong>Video 2</strong></div>
187
188 <div class="panel-body d-flex">
189 <div class="ytplayer" data-id="${VideoId2}" data-set="${Datatype2}"> </div>
190 <input class="videoDuration" name="v2d" type="hidden" />
191 <div>
192 <fieldset class="fileUploadFieldset"><label>First commentary<span class="timer">:</
        span></label><br />
193 <input class="audioinput fileUpload" single="" type="file" /><button class="
        audioinput record_button" disabled="disabled"><i class="material-icons">mic</i
        ></button>
194
195 <audio class="playback" controls="" disabled="disabled"> </audio>
196 <input class="duration" data-cid="1" name="v2c1d" type="hidden" /><input class="
        filekey" data-cid="1" name="v2c1" type="hidden" />
197 <div class="status"> </div>
198 Did any ads play while the video was playing?<br />
199 <input checked="checked" name="v2c1cm" required="" type="radio" value="0" /> No &
        nbsp;<input name="v2c1cm" type="radio" value="1" /> Yes</fieldset>
```

```
200
201 <fieldset class="fileUploadFieldset"><label>Second commentary<span class="timer
    ">:</span></label><br />
202 <input class="audioinput fileUpload" single="" type="file" /><button class="
    audioinput record_button" disabled="disabled"><i class="material-icons">mic</i
    ></button>
203
204 <audio class="playback" controls="" disabled="disabled"> </audio>
205 <input class="duration" data-cid="2" name="v2c2d" type="hidden" /><input class="
    filekey" data-cid="2" name="v2c2" type="hidden" />
206 <div class="status"> </div>
207 Did any ads play while the video was playing?<br />
208 <input checked="checked" name="v2c2cm" required="" type="radio" value="0" /> No &
    nbsp;<input name="v2c2cm" type="radio" value="1" /> Yes</fieldset>
209 </div>
210 </div>
211
212 <div class="panel-body">
213 <p><b>Questionnaire about Video 2</b></p>
214
215 <ul>
216   <li☆>How difficult was it to record a live commentary with this video? (1:Easy &
        larr; &rarr; 5:Difficult)<br />
217   <input name="v2_difficulty" required="" type="radio" value="1" /> 1  <input
        name="v2_difficulty" type="radio" value="2" /> 2  <input name="
        v2_difficulty" type="radio" value="3" /> 3  <input name="v2_difficulty"
        type="radio" value="4" /> 4  <input name="v2_difficulty" type="radio"
        value="5" /> 5 </li>
218   <li>Why was it difficult?
219   <ul>
220     <li>Speed (1:Too slow&larr; &rarr;5:Too fast; Check <b>3</b> if you answered ☆
          as 1:Easy)<br />
221     <input name="v2_speed" required="" type="radio" value="1" />1  <input name
        ="v2_speed" type="radio" value="2" /> 2  <input name="v2_speed" type="
        radio" value="3" /> 3  <input name="v2_speed" type="radio" value="4"
        /> 4  <input name="v2_speed" type="radio" value="5" /> 5 </li>
222     <li>Event changing in each scene cuts (1:Too few to continue describing&larr; &
        rarr;5:Too many to describe in time; Check <b>3</b> if you answered ☆
          as 1:Easy)<br />
223     <input name="v2_event" required="" type="radio" value="1" />1  <input name
        ="v2_event" type="radio" value="2" /> 2  <input name="v2_event" type="
        radio" value="3" /> 3  <input name="v2_event" type="radio" value="4"
        /> 4  <input name="v2_event" type="radio" value="5" /> 5 </li>
224     <li>Video content (situation, object, event,...etc)<br />
225     <input name="v2_content" required="" type="radio" value="unknown" /> 
        Unknown  <input name="v2_content" type="radio" value="known" />&
        nbsp;Known  <input name="v2_content" required="" type="radio"
        value="familiar" /> Familiar  </li>
226     <li>
227     <p>Other<br\> <input class="form-control" id="v2_other" name="v2_other"
        placeholder="e.g. 'The screen shakes so much that it is difficult to grasp
        the situation'" type="text" /> </br\></p>
228     </li>
229   </ul>
230   </li>
231   <li>How many scene cuts are there in this video?<br />
232   <input name="v2_scenecuts " required="" type="radio" value="1" /> 1  <input
        name="v2_scenecuts " type="radio" value="2-4" /> 2-4  <input name="
        v2_scenecuts " type="radio" value="5-8" /> 5-8  <input name="
        v2_scenecuts " type="radio" value="9-14" /> 9-14  <input name="
        v2_scenecuts " type="radio" value="over15" /> over 15- </li>
233 </ul>
234 </div>
```

```
</div>
</fieldset>

<fieldset><label>Questionnaire</label>

<div class="panel panel-primary">
<div class="panel-heading"><strong>Questionnaire about your background </strong></
    div>

<div class="panel-body">
<p>The country you are living in<br\> <input class="form-control" id="livingplace"
    name="livingplace" placeholder="e.g. UK" required="" type="text" /> </br\></p>

<p>The language(s) you can speak fluently<br\> <input class="form-control" id="
    language" name="language" placeholder="e.g. English, German,...etc" required=""
     type="text" /> </br\></p>

<p>Your familiarity with:</p>

<ul>
  <li>Sports: football, basketball, skating, ...<br\> <input class="form-control"
      id="sports" name="sports" placeholder="football, cricket,...etc or NONE"
      required="" type="text" /> </br\></li>
  <li>Music (instrument, genre): guitar, piano, rock, jazz, ...<br\> <input class="
      form-control" id="music" name="music" placeholder="violin, classical,...etc
      or NONE" required="" type="text" /></br\></li>
  <li>Other <br\> <input class="form-control" id="other" name="other" placeholder="
      computer science, video game,...etc or NONE" required="" type="text" /></br
      \></li>
</ul>
</div>
</div>
</fieldset>
</section>
<script>
  const disableMicrophone = false;

  const config = {
    region: 'ap-northeast-1',
    pool: 'ap-northeast-1:eaad08ff-2455-4c54-a19b-0cd9b21347f7',
    bucket: 'nedo3-amt-worker-uploads'
  }

  AWS.config.region = config.region;
  // Configure the credentials provider to use your identity pool
  AWS.config.credentials = new AWS.CognitoIdentityCredentials({
      IdentityPoolId: config.pool,
      RoleArn: 'arn:aws:iam::646640374441:role/
          Cognito_AMT_upload_files_poolUnauth_Role',
      AccountId: '646640374441',
  });




  // XXX DEBUG - for testing outside AMT HIT without actual uploaing
  if (!window.turkGetParam) {
    window.turkGetParam = () => 0;
    window.AWS = {
      S3: function() {
        return {
          upload: function(x, f) {
            console.log("[fake] uploading:", x)
```

```
            f(null, "");
          },
          getObject: function(x, f) {
            console.log("[fake] downloading:", x)
            f(null, {
              ContentType: 'audio/mp3',
              Data: new Uint8Array(),
            });
          },
        };
      },
      config: {},
      CognitoIdentityCredentials: function() {},
    };
  }

  const ytplayers = {};
  window.onYouTubeIframeAPIReady = function() {
    $('.ytplayer').each(function(index, div) {
      const $player = $(div);
      const videoId = $player.data('id');
      const player = new YT.Player(div, {
        width: 640,
        height: 390,
        videoId: videoId,
        events: {
          onReady: ytReady,
          onStateChange: ytStateChange,
        },
      });
      ytplayers[videoId] = player;
    });
  }

  const ytTag = document.createElement('script');
  ytTag.src = "https://www.youtube.com/iframe_api";
  const firstScriptTag = document.getElementsByTagName('script')[0];
  firstScriptTag.parentNode.insertBefore(ytTag, firstScriptTag);


  function ytReady(evt) {
    evt.target.mute();
    $('.audioinput.record_button').prop('disabled', false);
  }

  let $currentRecButton = null;
  let currentPlayer = null;
  let stopRecTimer = null;
  let timerUpdateTimer = null;
  let recStart = null;
  let videoDuration = null;

  function ytStateChange(evt) {
    if (evt.data == YT.PlayerState.ENDED) {
      videoDuration = evt.target.getDuration();
      if ($currentRecButton) {
        $currentRecButton.prop('disabled', false);
        stopRecTimer = setTimeout(function() {
          $currentRecButton.click();
        }, 60 * 1000);
      }
    }
  }
```

```
350
351   function supportsDynamicImport() {
352     try {
353       new Function('import("")');
354       return true;
355     } catch (err) {
356       return false;
357     }
358   }
359
360   function minsec(time) {
361     const min = Math.floor(time / 60);
362     const sec = Math.round(time - min * 60);
363     const filler = sec > 9 ? '' : '0';
364     return min + ":" + filler + sec;
365   }
366
367   function updateTimer() {
368     const duration = Math.round(currentPlayer.getDuration());
369     const position = Math.round((new Date() - recStart) / 1000);
370     const displayText = " (recorded " + minsec(position) + " / " + minsec(duration)
          + ")";
371     $currentRecButton.closest('fieldset').find('.timer').text(displayText);
372   }
373
374   const recorderPromise = new Promise(function(resolve, reject) {
375     if (disableMicrophone || !navigator.permissions) {
376       reject();
377     }
378
379     navigator.permissions.query({ name: 'microphone' })
380     .then(function(permissions) {
381       const micStatus = permissions.state;
382       if (micStatus == 'prompt' || micStatus == 'granted') {
383         if (supportsDynamicImport()) {
384           const vmsgURL = "https://cdn.jsdelivr.net/npm/vmsg@0.3.6/"
385           import(vmsgURL + "vmsg.js")
386           .then(function(vmsg) {
387             if (vmsg && vmsg.Recorder) {
388               const Recorder = vmsg.Recorder.bind(null, {
389                 wasmURL: vmsgURL + "vmsg.wasm",
390                 shimURL: vmsgURL + "wasm-polyfill.js",
391               });
392               resolve(Recorder);
393             } else {
394               reject();
395             }
396           });
397         } else {
398           reject();
399         }
400       } else {
401         reject();
402       }
403     });
404   });
405
406   const s3 = new AWS.S3({
407     apiVersion: "2006-03-01",
408     params: { Bucket: config.bucket }
409   });
410
411   const workerId = turkGetParam("workerId");
```

```
412   const assignmentId = turkGetParam("assignmentId");
413   const hitId = turkGetParam("hitId");
414
415   $('audio[data-loc]').each(function() {
416     const $this = $(this);
417     const fileKey = $this.data('loc');
418
419     s3.getObject(
420       {
421         Key: fileKey,
422       },
423       function(err, data) {
424         if (err) {
425           console.error(err)
426         } else {
427           const mimeType = data.ContentType;
428           const buffer = data.Body.buffer;
429           const blob = new Blob([buffer], {type: mimeType})
430           const audioURL = URL.createObjectURL(blob);
431           $('<source>').attr({
432             src: audioURL,
433             type: mimeType,
434           }).appendTo($this);
435           $this.prop('disabled', false);
436         }
437       }
438     );
439   });
440
441   function upload_file(file, videoId, commentaryId, videoSet) {
442     const extension = mimeTypes[file.type];
443     const fileKey = hitId + "_" + assignmentId + "_" + workerId + "_" + videoId + "
          _" + commentaryId + "_" + videoSet + "." + extension;
444
445     return new Promise(function(resolve, reject) {
446       s3.upload(
447         {
448           Key: fileKey,
449           Body: file,
450           ContentType: file.type,
451           ACL: "bucket-owner-full-control"
452         },
453         function(err, data) {
454           if (err) {
455             reject(err);
456           } else {
457             resolve(fileKey);
458           }
459         }
460       );
461     })
462   }
463
464   function do_input_upload(evt) {
465     const $fieldset = $(evt.target).closest("fieldset");
466     const $fileinput = $fieldset.find(".fileUpload");
467     const $status = $fieldset.find(".status");
468
469     const file = $fileinput.prop("files")[0];
470     if (!file) {
471       $status.text("");
472       return;
473     }
```

```
474       const mimeType = file.type;
475       const extension = mimeTypes[mimeType];
476
477       if (extension) {
478         receiveAudio(file, $fieldset);
479       } else {
480         $status.text("Please only upload MP3 or M4A files.").addClass('text-error').
              removeClass('text-success');
481       }
482     };
483
484     $(".fileUpload").on("input", do_input_upload);
485     $(function() {
486       $("#submitButton").prop("disabled", true);
487     });
488
489     const mimeTypes = {
490       'audio/webm;codecs=opus': 'webm',
491       'audio/ogg;codecs=opus': 'ogg',
492       'audio/mpeg': 'mp3',
493       'audio/x-m4a': 'm4a',
494       // TODO check uploaded mimetypes on other browsers (above is for Chrome)
495     };
496
497     function receiveAudio(blob, $fieldset) {
498       const $status = $fieldset.find(".status");
499       const $filekey = $fieldset.find(".filekey");
500       const $duration = $fieldset.find(".duration");
501       const $audioinputs = $fieldset.find(".audioinput");
502       const commentaryId = $filekey.data("cid");
503       const $player = $fieldset.closest(".panel-body").find(".ytplayer");
504       const $videoDuration = $fieldset.closest(".panel-body").find(".videoDuration");
505       const videoId = $player.data("id");
506       const videoSet = $player.data("set");
507
508       $status.text("Uploading...");
509       $audioinputs.prop("disabled", true);
510
511       const audioURL = URL.createObjectURL(blob);
512       const $audio = $fieldset.find('.playback').empty().prop('disabled', false);
513       const audio = $audio[0];
514       $('<source>').attr({
515         src: audioURL,
516         type: blob.type,
517       }).appendTo($audio);
518
519       const displayError = function(error) {
520         $status.text(error).addClass('text-danger').removeClass('text-success');
521         $filekey.val("");
522         $audioinputs.prop("disabled", false);
523       }
524
525       audio.onloadedmetadata = function audioLoadedMetadata() {
526         const duration = audio.duration;
527         if (videoDuration === null) {
528           displayError("Please watch the video.");
529           return;
530         }
531         if (duration < videoDuration) {
532           displayError("Please commentate the entire video.");
533           return;
534         }
535         $duration.val(duration);
```

```
536         $videoDuration.val(videoDuration);
537
538         upload_file(blob, videoId, commentaryId, videoSet)
539         .then(function(fileKey) {
540           $status.text("Success.").removeClass('text-danger').addClass('text-success
                ');
541           $filekey.val(fileKey);
542           $audioinputs.prop("disabled", false);
543           const $noComment = $('.fileUploadFieldset .filekey:not([value])');
544           if (!$noComment.length) {
545             // all comments uploaded, allow form submission
546             $("#submitButton").prop("disabled", false);
547           }
548         })
549         .catch(function() {
550           displayError("Failed to upload.");
551         });
552       };
553
554       audio.onerror = function audioError(evt) {
555         displayError("Audio file not valid.");
556       }
557
558       audio.load();
559     }
560
561     function getPlayerForSection(evt) {
562       const $audio = $(evt.target);
563       const videoId = $audio.closest('.panel-body').find('.ytplayer').data('id');
564       return ytplayers[videoId];
565     }
566
567     function onAudioPlay(evt) {
568       evt.target.ignorePause = true;
569       $('audio').not(evt.target).each(function() {
570         this.ignorePause = true;
571         this.pause();
572       });
573       ignorePause = false;
574       const time = evt.target.currentTime;
575       const player = getPlayerForSection(evt);
576       player.seekTo(time);
577       $currentRecButton = null;
578       videoDuration = null;
579       player.playVideo();
580     }
581
582     function onAudioPause(evt) {
583       if (evt.target.ignorePause) {
584         delete evt.target.ignorePause;
585         return;
586       }
587       const player = getPlayerForSection(evt);
588       player.pauseVideo();
589     }
590
591     function enableRecording(flag) {
592       if (flag !== false) {
593         $('.ytplayer').addClass('nocontrol');
594         $('.fileUploadFieldset .fileupload, .uploadway').hide();
595         $('.fileUploadFieldset .record_button, .recordway').show();
596         $('#audiomethod').val('recorded');
597       } else {
```

```
598         $('.ytplayer').removeClass('nocontrol');
599         $('.fileUploadFieldset .fileupload, .uploadway').show();
600         $('.fileUploadFieldset .record_button, .recordway').hide();
601         $('#audiomethod').val('uploaded');
602       }
603     }
604
605     $('audio')
606     .on('play', onAudioPlay)
607     .on('pause', onAudioPause);
608
609     recorderPromise
610     .then(function(Recorder) {
611       let recorder;
612
613       enableRecording();
614
615       function do_record(evt) {
616         const $recButton = $(evt.target).closest('button');
617         const $fieldset = $(evt.target).closest('.fileUploadFieldset');
618         const videoId = $fieldset.closest('.panel-body').find('.ytplayer').data('id')
              ;
619         currentPlayer = ytplayers[videoId];
620         const recording = $fieldset.data('recording');
621         if (recorder) {
622           recorder.stopRecording()
623           .then(function(blob) {
624             clearTimeout(stopRecTimer);
625             stopRecTimer = null;
626             clearTimeout(timerUpdateTimer);
627             timerUpdateTimer = null;
628             updateTimer();
629
630             currentPlayer.stopVideo();
631             receiveAudio(blob, $fieldset);
632             $recButton.html('<i class="material-icons">mic</i>');
633             $('.fileUploadFieldset').not($fieldset).find('.record_button').prop('
                  disabled', false);
634             recorder = null;
635           });
636         } else {
637           stopRecTimer = null;
638
639           $recButton.html('<i class="material-icons">stop</i>');
640           $currentRecButton = $recButton;
641           $('.fileUploadFieldset').find('.record_button').prop('disabled', true);
642           const chunks = [];
643           recorder = new Recorder();
644           recorder.initAudio()
645           .then(function() { return recorder.initWorker() })
646           .then(function() {
647             currentPlayer.playVideo();
648             recStart = new Date();
649             videoDuration = null;
650             timerUpdateTimer = setInterval(updateTimer, 1000);
651             return recorder.startRecording();
652           })
653           .catch(function(...args) {
654             enableRecording(false);
655           });
656         }
657         return false;
658       }
```

```
    $(".record_button").on("click", do_record);
  })
</script>
```

### B.2.2 コメント修正 UI

Listing B.5: Transcribe

```
<script src="https://sdk.amazonaws.com/js/aws-sdk-2.142.0.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></
    script>
<link href="https://s3.amazonaws.com/mturk-public/bs30/css/bootstrap.min.css" rel="
    stylesheet" />
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="
    stylesheet" />
<style type="text/css">
  /* https://css-tricks.com/auto-growing-inputs-textareas/ */
  .input-sizer {
    display: inline-grid;
    vertical-align: top;
    align-items: center;
    position: relative;
    /* border: solid 1px; */
    /* padding: 0.25em 0.5em; */
    margin: 5px;
  }
  .input-sizer.stacked {
    /* padding: 0.5em; */
    align-items: stretch;
  }
  .input-sizer.stacked::after,
  .input-sizer.stacked input,
  .input-sizer.stacked textarea {
    grid-area: 2/1;
  }
  .input-sizer::after,
  .input-sizer input,
  .input-sizer textarea {
    width: auto;
    min-width: 1em;
    grid-area: 1/2;
    font: inherit;
    padding: 0.25em;
    margin: 0;
    resize: none;
    background: none;
    appearance: none;
    border: none;
  }
  .input-sizer span {
    padding: 0.25em;
  }
  .input-sizer::after {
    content: attr(data-value) " ";
    visibility: hidden;
    white-space: pre-wrap;
  }
  /*
```

```
  .input-sizer:focus-within {
    outline: solid 1px blue;
    box-shadow: 4px 4px 0px blue;
  }
  .input-sizer:focus-within > span {
    color: blue;
  }
  */
  .input-sizer:focus-within textarea:focus,
  .input-sizer:focus-within input:focus {
    outline: none;
  }
  /* end input-sizer */

  .players {
    position: sticky;
    top: 20px;
    padding-right: 20px;
  }

  .ytplayer {
    pointer-events: none;
  }

  audio {
    outline: none;
  }

  .no-bold-labels label {
    font-weight: normal;
  }

  .d-flex {
    display: flex;
  }

  .align-items-start {
    align-items: start;
  }

  .flex-row {
    flex-direction: row;
  }

  .flex-grow-1 {
    flex-grow: 1;
  }

  blockquote.example {
    border: 0;
    padding: 0;
    margin: 0;
  }
  blockquote.example:before {
    content: "e.g. ";
    color: #bbbbbb;
  }

  fieldset {
    padding: 10px;
    background: #fbfbfb;
    border-radius: 5px;
    margin-bottom: 5px;
```

```
111   }
112
113   .material-icons.smaller {
114     margin: -3px;
115   }
116
117   .align-self-center {
118     align-self: center;
119   }
120
121   .form-check-inline {
122     display: inline-block;
123     margin-right: 10px;
124   }
125
126   .w-100 {
127     width: 100%;
128   }
129
130   .mb-10 {
131     margin-bottom: 10px;
132   }
133
134   .mb-20 {
135     margin-bottom: 20px;
136   }
137
138   textarea:required:invalid {
139     border: 2px solid #ff0000;
140   }
141
142   fieldset.required {
143     border: 2px solid transparent;
144   }
145
146   fieldset.required:invalid {
147     border-bottom: 2px solid #ff0000;
148     border-radius: 0;
149   }
150
151   form:invalid input[type="submit"] {
152     pointer-events: none;
153     opacity: 0.38;
154   }
155
156   kbd {
157     font-weight: bold;
158   }
159 </style>
160 <main class="container" id="Other" style="margin-bottom:15px; padding: 10px 10px;
        font-family: Verdana, Geneva, sans-serif; color:#333333; font-size:0.9em;">
161   <fieldset><label>Read the Instructions</label>
162     <div class="panel panel-primary">
163       <div class="panel-heading"><strong>Instructions</strong></div>
164
165       <div class="panel-body">
166         <!-- <p>This is a qualification test for the task explained below.</p> -->
167         <!-- <p>You can accept the actual HIT after you pass this qualification
              task!</p> -->
168         <div>This HIT is a task of listening to audio and correcting the
              transcription.</div>
169
170         <div>There are two audios, so correct those transcriptions.</div>
```

```
171
172         <!-- <div>If you do not follow the rules below,your job will be rejected.</
                div> -->
173         <div>Please follow the instructions below to do both of the following <b>"
                Transcription correction"</b> and <b>"Transcription review"</b>.
174
175         <div>Please check your work before submission, you cannot redo this HIT
                after submission.</div>
176         <div>The reward of this task is determined by the number of chars in the
                pre-transcription and the audio time.</div>
177
178         <h4>Transcription correction fields</h4>
179         <ul>
180           <li>Please add an appropriate punctuation mark (".", "?" etc.) at the end
                , if you find the sentence is ending.</li>
181
182           <li>Please input <b>"(UNKNOWN)"</b> in places where you can't hear or
                recognise what was said.</li>
183
184           <li>Please <b>correct</b> any part where the automatic transcription text
                 differs from the audio.
185             <blockquote class="example">
186               "So it z there a burning man?" &rarr; "So is there a burning man?"
187             </blockquote>
188           </li>
189           <li>Please <b>write out</b> any <b>numbers</b> using <b>words</b> as it
                is.
190             <blockquote class="example">
191               "1" &rarr; "one"
192             </blockquote>
193             <blockquote class="example">
194               "1234" &rarr; "one thousand two hundred and thirty four"
195             </blockquote>
196             <blockquote class="example">
197               "1st,2nd" &rarr; "first,second"
198             </blockquote>
199
200           </li>
201         </ul>
202
203         <h4>Transcription review fields</h4>
204         <h5>In the transcription review fields, please perform all the corrections
                listed for the Transcription correction fields, as well as the
                following additional ones.</h5>
205         <ul>
206         <!--  <li>In the transcription review fields, please perform all the
                corrections listed for the Transcription correction fields, as well as
                the following additional ones.</li> -->
207           <li>Please <b>change</b> the <b>"(UNKNOWN)"</b> to appropriate word or <b
                >remove</b> for natural sentence.
208           </li>
209           <li>Please <b>correct</b> any incorrect grammar.
210             <blockquote class="example">
211               "They is talking." &rarr; "They are talking."
212             </blockquote>
213             <blockquote class="example">
214               "There are two mouses." &rarr; "There are two mice."
215             </blockquote>
216           </li>
217           <li>
218             Please <b>remove</b> rephrasing.
219             <blockquote class="example">
```

```
220               "and I think oh it seems one of them, he threw three darts," &rarr; "
                      He threw three darts."
221             </blockquote>
222           </li>
223           <li>
224             Please <b>remove</b> any representations of fillers* words or
                    hesitation markers.
225             <br/><b>*fillers: </b><a href="https://en.wikipedia.org/wiki/Filler_(
                    linguistics)" target="_blank">>https://en.wikipedia.org/wiki/
                    Filler_(linguistics)</A>
226             <blockquote class="example">
227               fillers: <i>Mmm,Uh, uh, OK, Yeah</i>...etc
228             </blockquote>
229
230             <blockquote class="example">
231               "uh Let's see uh there's a man, there's a man throwing darts." &rarr;
                      "There's a man throwing darts."
232             </blockquote>
233             <blockquote class="example">
234               "Mmm,is there a burning man?" &rarr; "Is there a burning man?"
235             </blockquote>
236           </li>
237
238           <li>If the field ends up as empty after such deletions, please input <b
                  >"(NONE)"</b>.
239           </li>
240         </ul>
241         <section>
242           <fieldset class="required">
243             <label>
244               <input name="Check" required type="checkbox" value="check_ok" />
245               Have you checked the instructions?
246             </label>
247           </checkbox>
248         </section>
249       </div>
250     </div>
251   </fieldset>
252
253 <!-- (2022.02.02) -->
254
255   <fieldset class="task">
256     <label>Task</label>
257
258     <div class="panel panel-primary">
259       <div class="panel-heading"><strong>Audio Clip</strong></div>
260
261       <div class="panel-body d-flex align-items-start">
262         <div class="players" style="width: 480px; padding: 0">
263           <audio controls="controls" class="mb-20 w-100"></audio>
264           <div class="ytplayer mb-20"></div>
265           <table class="mb-10">
266             <tr>
267               <td><kbd class="modkey"></kbd><kbd>I</kbd>:</td>
268               <td>Play sentence from start</td>
269             </tr>
270             <tr>
271               <td><kbd class="modkey"></kbd><kbd>B</kbd>:</td>
272               <td>Rewind one second</td>
273             </tr>
274             <tr>
275               <td><kbd class="modkey"></kbd><kbd>U</kbd>:</td>
276               <td>Pause and unpause</td>
```

```
277             </tr>
278           </table>
279               <br/><b>*Instructions: </b><a href="https://material-analyzer.airc.
                      aist.go.jp/InstructionsHelp.html" target="_blank">>Help
280 </A><br/>
281            <small style="border-top: 0.5px solid #000000">(Only when cursor is in a
                    subtitle editing field)</small>
282         </div>
283
284         <div class="sentences flex-grow-1">
285           <!-- sentences go here -->
286         </div>
287       </div>
288       <section class="panel-body panel-primary">
289         <div class="panel-heading d-flex flex-row">
290           Questionnaire about this task
291         </div>
292         <div class="panel-body no-bold-labels">
293           <div>
294             Could you understand what was said?
295           </div>
296           <fieldset class="required">
297             <div class="form-check form-check-inline">
298               <input required class="form-check-input" type="radio" id="2
                      _understood_y" name="2_understood" value="1">
299               <label class="form-check-label" for="understood_y">Yes</label>
300             </div>
301             <div class="form-check form-check-inline">
302               <input required class="form-check-input" type="radio" id="2
                      _understood_n" name="2_understood" value="0">
303               <label class="form-check-label" for="understood_n">No</label>
304             </div>
305           </fieldset>
306           <div>
307             If not, why?
308           </div>
309           <fieldset>
310             <div class="form-check form-check-inline">
311               <input class="form-check-input" type="checkbox" id="2_low_volume"
                      name="2_problems" value="low_volume">
312               <label class="form-check-label" for="2_low_volume">Low volume</label>
313             </div>
314             <div class="form-check form-check-inline">
315               <input class="form-check-input" type="checkbox" id="2_too_fast" name
                      ="2_problems" value="too_fast">
316               <label class="form-check-label" for="2_too_fast">Too fast</label>
317             </div>
318             <div class="form-check form-check-inline">
319               <input class="form-check-input" type="checkbox" id="2_unknown_meaning
                      " name="2_problems" value="unknown_meaning">
320               <label class="form-check-label" for="2_unknown_meaning">Words of
                      unknown meaning</label>
321             </div>
322             <div class="form-check form-check-inline">
323               <input class="form-check-input" type="checkbox" id="2_not_english"
                      name="2_problems" value="not_english">
324               <label class="form-check-label" for="2_not_english">Not English</
                      label>
325             </div>
326             <div class="form-check form-check-inline">
327               <input class="form-check-input" type="checkbox" id="2_too_much_noise"
                       name="2_problems" value="too_much_noise">
```

```
328               <label class="form-check-label" for="2_too_much_noise">Too much noise
                      </label>
329             </div>
330             <div class="form-check form-check-inline">
331               <input class="form-check-input" type="checkbox" id="2_other" name="2
                      _problems" value="other">
332               <label class="form-check-label" for="2_other">Other</label>
333             </div>
334           </fieldset>
335           <div>
336             Did the audio match the video?
337           </div>
338           <fieldset class="required">
339             <div class="form-check form-check-inline">
340               <input required class="form-check-input" type="radio" id="2_matched_y
                      " name="2_matched" value="1">
341               <label class="form-check-label" for="matched_y">Yes</label>
342             </div>
343             <div class="form-check form-check-inline">
344               <input required class="form-check-input" type="radio" id="2_matched_n
                      " name="2_matched" value="0">
345               <label class="form-check-label" for="matched_n">No</label>
346             </div>
347           </fieldset>
348           <div>
349             Was there any inappropriate language?
350           </div>
351           <fieldset class="required">
352             <div class="form-check form-check-inline">
353               <input required class="form-check-input" type="radio" id="2
                      _appropriate_y" name="2_appropriate" value="0">
354               <label class="form-check-label" for="appropriate_y">Yes</label>
355             </div>
356             <div class="form-check form-check-inline">
357               <input required class="form-check-input" type="radio" id="2
                      _appropriate_n" name="2_appropriate" value="1">
358               <label class="form-check-label" for="appropriate_n">No</label>
359             </div>
360           </fieldset>
361         </div>
362       </section>
363     </div>
364   </fieldset>
365 </main>
366 <input name="resultJSON" id="resultJSON" type="hidden"/>
367
368
369 <script id="realdata" type="text/json">${json}</script>
370 <script id="sampledata" type="text/json">
371   {"sentences":["uh Let's see uh there's a man, there's a man throwing darts", "and
        I think oh it seems one of them, he threw three darts,", "and I think 11 of
      them hit the center, didn't it?."], "defiller_sentences":["Let's see... there
      's a man, there's a man throwing darts", "and I think... it seems one of them
      , he threw three darts,", "and I think 11 of them hit the center, didn't it
      ?."], "timestamps":[[0.54,6.35],[6.35,10.01],[10.01,14.364]], "audio_s3_key
      ":"sample/sample_with_errors.mp3"}
372 </script>
373 <script id="testdata" type="text/json">
374   {"sentences":["All right.","So it z there a burning man? It's a man and a woman
      .","Man has some bongos.","Um, the woman has incentive for and feeds braided
      hair with beads in it.","They're having some kind of, ah, conversation.","She
       wants to see him do something on his his bongos.","He's got 12345 bongos.","
      Another one comes in and taste the, uh uh, container.","Okay, so he's the man
```

```
       is playing on the bongos.","He got a bicyclist, uh, in an odd costume I was
      walking about.","She's dancing.","She's got a uh huh And tiny barely there in
       little skirt in a very tight little T shirt.","And she is now dancing to his
       his bongo playing.","And he has, ah, she's doing some unique moves.","Her
      her derriere is really hanging out of her out of her costume.","And behind
      them, there's, um, Kintz there in the desert.","Looks like that there got
      some tents behind them with a bunch of bicycles just lined up.","And she's
      just dancing away to hiss bongo playing.","And he looks like he has attempt
      behind him that is in the shape of a drum.","Bongo.","Possibly.","And she's
      clapping.","She's She's pleased with his bongo playing."],"defiller_sentences
      ":["All right.","So it z there a burning man? It's a man and a woman.","Man
      has some bongos.","The woman has incentive for and feeds braided hair with
      beads in it.","They're having some kind of... conversation.","She wants to
      see him do something on his his bongos.","He's got 12345 bongos.","Another
      one comes in and taste the... container.","Okay, so he's the man is playing
      on the bongos.","He got a bicyclist... in an odd costume I was walking about
      .","She's dancing.","She's got a...  And tiny barely there in little skirt in
       a very tight little T shirt.","And she is now dancing to his his bongo
      playing.","And he has... she's doing some unique moves.","Her her derriere is
       really hanging out of her out of her costume.","And behind them, there's...
      Kintz there in the desert.","Looks like that there got some tents behind them
       with a bunch of bicycles just lined up.","And she's just dancing away to
      hiss bongo playing.","And he looks like he has attempt behind him that is in
      the shape of a drum.","Bongo.","Possibly.","And she's clapping.","She's She's
       pleased with his bongo playing."],"timestamps
      ":[[2.34,2.82],[2.82,7.36],[7.36,8.96],[9.34,15.56],[16.54,18.85],[20.44,24.35],[24.74,26.75],[
      audio_s3_key":"Batch_4345895/Approved/3
      UDTAB6HID6XCUM4XL77KBQ5G7L90G_3C6FJU71TY0VG13MT6I8XJ3Q86UUYX_A44ODU03OP0D9_--
      veKG73Di4_1_train.mp3"}
</script>

<script>
  "use strict";

  const config = {
    region: 'ap-northeast-1',
    bucket: 'nedo3-amt-worker-uploads',
  };

  const realJson = $('#realdata').text();
  const testJson = $('#testdata').text();
  const sampleJson = $('#sampledata').text();
  const data = JSON.parse(realJson === '$' + '{json}' ? testJson : realJson);
  const sampleData = JSON.parse(sampleJson);

  const searchParams = (new URL(document.URL)).searchParams;
  const workerId = searchParams.get('workerId');
  const assignmentId = searchParams.get('assignmentId');

  const audioURL = "https://" + config.bucket + ".s3-" + config.region + ".
      amazonaws.com/" + data.audio_s3_key;
  const videoID = data.audio_s3_key.match(/_(.{11})_[^_]+_[^_]+\.mp3$/)[1];
  const sampleAudioURL = "https://" + config.bucket + ".s3-" + config.region + ".
      amazonaws.com/" + sampleData.audio_s3_key;
  const sampleVideoID = "H6hb3SIqdL4";


  const ytAPIPromise = new Promise((resolve, reject) => {
    window.onYouTubeIframeAPIReady = resolve;
    const ytTag = document.createElement('script');
    ytTag.src = "https://www.youtube.com/iframe_api";
    const firstScriptTag = document.getElementsByTagName('script')[0];
    firstScriptTag.parentNode.insertBefore(ytTag, firstScriptTag);
```

```
407   });
408
409   function makeYTPlayerPromise(wrapper, options) {
410     return ytAPIPromise.then(() =>
411       new Promise((resolve, reject) => {
412         const ytPlayer = new YT.Player(wrapper, {
413           ...options,
414           events: {
415             ...(options.events ?? {}),
416             onReady: () => {
417               ytPlayer.mute();
418               resolve(ytPlayer);
419             },
420           }
421         })
422         return ytPlayer;
423       })
424     )
425   }
426
427   let modKey;
428   if (navigator.platform === 'MacIntel') {
429     $('kbd.modkey').html('^^e2^^8c^^98')
430     modKey = 'metaKey';
431   } else {
432     $('kbd.modkey').text('Ctrl').after('-');
433     modKey = 'ctrlKey';
434   }
435   const keys = {
436     play: 'i',
437     back: 'b',
438     pause: 'u',
439     enter: 'Enter',
440   };
441   const backJump = 1; // seconds
442
443   $('form#mturk_form').submit(evt => {
444     const $invalid = $('main.container').find(':invalid');
445     if ($invalid.length) {
446       return false;
447     }
448
449     const sampleBoxes = {}, actualBoxes = {};
450     $('.task.sample input[type="radio"]:checked').get().forEach(el => sampleBoxes[
          el.name.substr(2)] = !!+el.value);
451     $('.task.sample input[type="checkbox"]').get().forEach(el => (sampleBoxes[el.
          name.substr(2)] ??= {})[el.value] = el.checked);
452     $('.task.actual input[type="radio"]:checked').get().forEach(el => actualBoxes[
          el.name.substr(2)] = !!+el.value);
453     $('.task.actual input[type="checkbox"]').get().forEach(el => (actualBoxes[el.
          name.substr(2)] ??= {})[el.value] = el.checked);
454     const result = {
455       actual: {
456         correction: $('.task.actual .input-correction').get().map(el => el.value),
457         review: $('.task.actual .input-review').get().map(el => el.value),
458         boxes: actualBoxes,
459       },
460       sample: {
461         correction: $('.task.sample .input-correction').get().map(el => el.value),
462         review: $('.task.sample .input-review').get().map(el => el.value),
463         boxes: sampleBoxes,
464       },
465     };
```

```
466       $('#resultJSON').val(JSON.stringify(result));
467       // no preventDefault
468     });
469
470
471
472     function makeTask($task, which, audioURL, videoID, data) {
473       function playSentence(evt, back=0) {
474         const $element = $(evt.target).closest('section');
475         const startTime = $element.data('from');
476         const newTime = back
477           ? Math.max(startTime, audio.currentTime - back)
478           : startTime;
479
480         audio.currentTime = newTime;
481         audio.play();
482       };
483
484       $task.addClass(which);
485
486       const ytPlayerPromise = makeYTPlayerPromise($task.find('.ytplayer')[0], {
487         width: 480,
488         height: 292,
489         videoId: videoID,
490         playerVars: {
491           autoplay: 0,
492           controls: 0,
493           disablekb: 1,
494           modestbranding: 1,
495           fs: 0,
496         },
497       });
498
499       const $audio = $task.find('audio');
500       const audio = $audio[0];
501       console.log(audio);
502       const audioPromise = new Promise((resolve, reject) => {
503         audio.addEventListener('canplaythrough', () => resolve(audio));
504         $audio.attr('src', audioURL);
505       });
506
507       Promise.all([ytPlayerPromise, audioPromise]).then(([ytPlayer, audio]) => {
508
509         audio.addEventListener('seeked', evt => {
510           ytPlayer.seekTo(audio.currentTime);
511           if (audio.paused) {
512             ytPlayer.pauseVideo();
513           } else {
514             ytPlayer.playVideo();
515           }
516         });
517
518         audio.addEventListener('pause', evt => {
519           ytPlayer.pauseVideo();
520         });
521
522         audio.addEventListener('play', evt => {
523           ytPlayer.playVideo();
524         });
525
526         const $sentences = $task.find('.sentences');
527         $sentences
528         .on('click', '.reset-button', (evt) => {
```

```
529           const $input = $(evt.target).closest('.sentbox').find('textarea');
530           $input.val($input.data('original'));
531         })
532         .on('click', '.play-button', playSentence)
533         .on('keydown', 'textarea', (evt) => {
534           let handled = false;
535           if (evt.key == keys.enter) {
536             evt.preventDefault();
537             return;
538           }
539           if (evt[modKey]) {
540             if (evt.key == keys.play) {
541               playSentence(evt);
542               handled = true;
543             } else if (evt.key == keys.back) {
544               playSentence(evt, backJump);
545               handled = true;
546             } else if (evt.key == keys.pause) {
547               if (audio.paused) {
548                 audio.play();
549               } else {
550                 audio.pause();
551               }
552               handled = true;
553             }
554           }
555           if (handled) {
556             evt.stopPropagation();
557             evt.preventDefault();
558           }
559         });
560
561         data.timestamps.forEach(([from, to], i) => {
562           const sentence = data.sentences[i];
563           const defillerSentence = data.defiller_sentences[i];
564           const $template = $('<div/>').html(`
565               <section class="panel-body panel-primary">
566                 <div class="panel-heading d-flex flex-row">
567                   <div class="align-self-center">
568                     <span class="from"></span>s &mdash; <span class="to"></
                          span>s
569                   </div>
570                   <div class="flex-grow-1"></div>
571                   <i class="play-button smaller material-icons align-self-center btn
                        ">play_arrow</i>
572                 </div>
573                 <div class="panel-body">
574                   <div class="w-100 sentbox">
575                     <label>Transcription correction</label>
576                     <div class="d-flex flex-row">
577                       <div class="input-sizer stacked flex-grow-1 align-self-start">
578                         <textarea class="input-correction form-control w-100 h-100"
                              rows="1" value="" required /></textarea>
579                       </div>
580                       <i class="reset-button material-icons align-self-start btn">
                            replay</i>
581                     </div>
582                   </div>
583                   <br/>
584                   <div class="w-100 sentbox">
585                     <label>Transcription review:</label>
586                     <div class="d-flex flex-row">
587                       <div class="input-sizer stacked flex-grow-1 align-self-start">
```

```
                        <textarea class="input-review form-control w-100 h-100" rows
                            ="1" value="" required /></textarea>
                    </div>
                    <i class="reset-button material-icons align-self-start btn">
                        replay</i>
                  </div>
                </div>
              </div>
            </section>
        `);
        const $element = $template.children(':first');
        $sentences.append($element);
        $element.data('from', from);
        $element.find('.from').text(from);
        $element.find('.to').text(to);
        $element.find('.input-correction').val(sentence).data('original', sentence)
            .parent().attr('data-value', sentence);
        $element.find('.input-review').attr('placeholder', defillerSentence).data('
            original', '').parent().attr('data-value', defillerSentence);
        $element.find('.input-sizer > textarea').on('input', evt => {
          evt.target.parentNode.dataset.value = evt.target.value || evt.target.
              placeholder;
        });
      });

    });
  }
  const task_els = $('.task').get();
  <!-- makeTask($(task_els[0]), 'sample', sampleAudioURL, sampleVideoID, sampleData
      ); -->
  <!-- makeTask($(task_els[1]), 'actual', audioURL, videoID, data); -->
  makeTask($(task_els[0]), 'actual', audioURL, videoID, data);
</script>
```

## B.3 概況テキスト生成

Listing B.6: Recording

```
import argparse
import warnings
from datetime import datetime
from functools import reduce
from pathlib import Path

import torch


from network_attention import (
    AttnDecoder as Decoder,
    Encoder_nonHigh as Encoder,
    EncoderDecoder,
    setup_attention
```

```
)
from train import run
from postprocessing.bleu import calc_bleu
from postprocessing.export import export_results_to_csv
from preprocessing.dataset import create_dataset
from util.config import Config
from util.constant import Phase, SpecialToken, SeqType
from util.logging import create_logger
from preprocessing.prepro import prepro


def parse_args() -> argparse.Namespace:

    parser = argparse.ArgumentParser(prog='reporter')
    parser.add_argument('--device',
                        type=str,
                        metavar='DEVICE',
                        default='cpu',
                        help='`cuda:n` where `n` is an integer, or `cpu`')
    parser.add_argument('--debug',
                        dest='is_debug',
                        action='store_true',
                        default=False,
                        help='show detailed messages while execution')
    parser.add_argument('--config',
                        type=str,
                        dest='dest_config',
                        metavar='FILENAME',
                        default='config.toml',
                        help='specify config file (default: `config.toml`)')
    parser.add_argument('-m',
                        '--model',
                        type=str,
                        metavar='FILENAME')
    parser.add_argument('-o',
                        '--output-subdir',
                        type=str,
                        metavar='DIRNAME')
    return parser.parse_args()


def main() -> None:

    args = parse_args()

    if not args.is_debug:
        warnings.simplefilter(action='ignore', category=FutureWarning)

    config = Config(args.dest_config)

    device = torch.device(args.device)
    """
    now = datetime.today().strftime('reporter-%Y-%m-%d-%H-%M-%S')
    dest_dir = config.dir_output / Path(now) \
        if args.output_subdir is None \
        else config.dir_output / Path(args.output_subdir)
    """
    now = datetime.today().strftime('reporter-%Y-%m-%d-%H-%M-%S')
    folder_name = "test"
    dest_dir = config.dir_output / Path(folder_name) \
        if args.output_subdir is None \
        else config.dir_output / Path(args.output_subdir)
    dest_log = dest_dir / Path(now + '_' + 'reporter.log')
```

```
78
79      """
80      now = "reporter-2019-09-11-11-36-21"
81      dest_dir = config.dir_output / Path(now) \
82          if args.output_subdir is None \
83          else config.dir_output / Path(args.output_subdir)
84      """
85
86      #dest_log = dest_dir / Path('reporter.log')
87
88      logger = create_logger(dest_log, is_debug=args.is_debug)
89      config.write_log(logger)
90
91      message = 'start main (is_debug: {}, device: {})'.format(args.is_debug, args.
            device)
92      logger.info(message)
93
94      # === Alignment ===
95      has_all_alignments = \
96          reduce(lambda x, y: x and y,
97                 [(config.dir_output / Path('alignment-{}.json'.format(phase.value)))
                       .exists()
98                  for phase in list(Phase)])
99
100     # がなければ，作るalignment
101     if not has_all_alignments:
102         print("not exist alignment file")
103         prepro(config, logger)
104
105     # === Dataset ===
106     (vocab, train, valid, test) = create_dataset(config, device)
107
108     vocab_size = len(vocab)
109     dest_vocab = dest_dir / Path('reporter.vocab')
110     with dest_vocab.open(mode='wb') as f:
111         torch.save(vocab, f)
112     seqtypes = [SeqType.NormMovRefLong,
113                           SeqType.NormMovRefShort,
114                           SeqType.StdLong,
115                           SeqType.StdShort] \
116         if config.use_standardization \
117         else [SeqType.NormMovRefLong,
118               SeqType.NormMovRefShort]
119     attn = setup_attention(config, seqtypes)
120     encoder = Encoder(config, device)
121     decoder = Decoder(config, vocab_size, attn, device)
122     model = EncoderDecoder(encoder, decoder, device)
123     optimizer = torch.optim.Adam(model.parameters(), lr=config.learning_rate)
124     criterion = torch.nn.NLLLoss(reduction='elementwise_mean',
125                                  ignore_index=vocab.stoi[SpecialToken.Padding.value
                                         ])
126
127     # === Train ===
128     dest_model = dest_dir / Path('reporter.model')
129     prev_valid_bleu = 0.0
130     max_bleu = 0.0
131     best_epoch = 0
132     early_stop_counter = 0
133     for epoch in range(config.n_epochs):
134         logger.info('start epoch {}'.format(epoch))
135         train_result = run(train,
136                            vocab,
137                            model,
```

```
                              optimizer,
                              criterion,
                              Phase.Train,
                              logger, "")
        train_bleu = calc_bleu(train_result.gold_sents, train_result.pred_sents)
        valid_result = run(valid,
                              vocab,
                              model,
                              optimizer,
                              criterion,
                              Phase.Valid,
                              logger, "")
        valid_bleu = calc_bleu(valid_result.gold_sents, valid_result.pred_sents)

        s = ' | '.join(['epoch: {0:4d}'.format(epoch),
                        'training loss: {:.2f}'.format(train_result.loss),
                        'training BLEU: {:.4f}'.format(train_bleu),
                        'validation loss: {:.2f}'.format(valid_result.loss),
                        'validation BLEU: {:.4f}'.format(valid_bleu)])
        logger.info(s)

        if max_bleu < valid_bleu:
            torch.save(model.state_dict(), str(dest_model))
            max_bleu = valid_bleu
            best_epoch = epoch

        early_stop_counter = early_stop_counter + 1 \
            if prev_valid_bleu > valid_bleu \
            else 0
        if early_stop_counter == config.patience:
            logger.info('EARLY STOPPING')
            break
        prev_valid_bleu = valid_bleu

    # === Test ===
    with dest_model.open(mode='rb') as f:
        model.load_state_dict(torch.load(f))
    test_result = run(test,
                      vocab,
                      model,
                      optimizer,
                      criterion,
                      Phase.Test,
                      logger, dest_dir)
    test_bleu = calc_bleu(test_result.gold_sents, test_result.pred_sents)

    s = ' | '.join(['epoch: {:04d}'.format(best_epoch),
                    'Test Loss: {:.2f}'.format(test_result.loss),
                    'Test BLEU: {:.10f}'.format(test_bleu)])
    logger.info(s)

    export_results_to_csv(dest_dir, test_result)


if __name__ == '__main__':
    main()
```

### B.3.1 訓練プログラム

Listing B.7: Transcribe

```
from logging import Logger
from typing import Dict, List

import numpy
import torch
from nltk.translate.bleu_score import SmoothingFunction, sentence_bleu
from torchtext.data import Iterator
from torchtext.vocab import Vocab

from network_attention import Attention, EncoderDecoder
from operation import (
    get_latest_closing_vals,
    replace_tags_with_vals
)
from postprocessing.text import remove_bos
from util.constant import SEED, Code, Phase, SeqType, SpecialToken
from util.conversion import stringify_ric_seqtype
from util.tool import takeuntil
from pathlib import Path


class RunResult:
    def __init__(self,
                 loss: float,
                 article_ids: List[str],
                 gold_sents: List[List[str]],
                 gold_sents_num: List[List[str]],
                 pred_sents: List[List[str]],
                 pred_sents_num: List[List[str]]):

        self.loss = loss
        self.article_ids = article_ids
        self.gold_sents = gold_sents
        self.gold_sents_num = gold_sents_num
        self.pred_sents = pred_sents
        self.pred_sents_num = pred_sents_num


import matplotlib.pyplot as plt
import japanize_matplotlib
plt.switch_backend('agg')
import matplotlib.ticker as ticker


def showAttention(input, output_words, attentions, save_file):
    # Set up figure with colorbar
    fig = plt.figure(figsize=(60, 40))
    ax = fig.add_subplot(111)
    cax = ax.matshow(attentions, cmap='bone')
    fig.colorbar(cax)

    # Set up axes
    #print(input)
    #print(input.cpu())
    ax.set_xticklabels(['']  + [str(round(s, 3)) for s in input.cpu().numpy().tolist
        ()] + [''], rotation=90)
    #ax.set_xticklabels(['']  + input.split(' ') +['<EOS>'], rotation=90)
    ax.set_yticklabels(['']  + output_words)

    # Show label at every tick
    ax.xaxis.set_major_locator(ticker.MultipleLocator(1))
```

```
61      ax.yaxis.set_major_locator(ticker.MultipleLocator(1))
62
63      plt.savefig(save_file)
64      plt.close()
65
66
67  def run(X: Iterator,
68          vocab: Vocab,
69          model: EncoderDecoder,
70          optimizer: Dict[SeqType, torch.optim.Optimizer],
71          criterion: torch.nn.modules.Module,
72          phase: Phase,
73          logger: Logger,
74          dest_dir) -> RunResult:
75
76      if phase in [Phase.Valid, Phase.Test]:
77          model.eval()
78      else:
79          model.train()
80
81      numpy.random.seed(SEED)
82
83      accum_loss = 0.0
84      all_article_ids = []
85      all_gold_sents = []
86      all_pred_sents = []
87      all_gold_sents_with_number = []
88      all_pred_sents_with_number = []
89      attn_weights = []
90
91      for batch in X:
92
93          article_ids = batch.article_id
94          times = batch.time
95          tokens = batch.token
96          raw_short_field = stringify_ric_seqtype(Code.N225.value, SeqType.RawShort)
97          latest_vals = [x for x in getattr(batch, raw_short_field).data[:, 0]]
98          raw_long_field = stringify_ric_seqtype(Code.N225.value, SeqType.RawLong)
99          latest_closing_vals = get_latest_closing_vals(batch, raw_long_field, times)
100         max_n_tokens, _ = tokens.size()
101
102         # Forward
103         if phase == Phase.Test:
104             loss, pred, attn_weight, input_data = model(batch, batch.batch_size,
                    tokens, times, criterion, phase)
105         else:
106             loss, pred, attn_weight = model(batch, batch.batch_size, tokens, times,
                     criterion, phase)
107
108         if phase == Phase.Train:
109             optimizer.zero_grad()
110             loss.backward()
111             optimizer.step()
112
113         if isinstance(model.decoder.attn, Attention):
114             attn_weight = numpy.array(list(zip(*attn_weight)))
115             attn_weights.extend(attn_weight)
116
117         all_article_ids.extend(article_ids)
118
119         i_eos = vocab.stoi[SpecialToken.EOS.value]
120         # Recover words from ids removing BOS and EOS from gold sentences for
              evaluation
```

```
121         gold_sents = [remove_bos([vocab.itos[i] for i in takeuntil(i_eos, sent)])
122                       for sent in zip(*tokens.cpu().numpy())]
123         all_gold_sents.extend(gold_sents)
124 
125         pred_sents = [remove_bos([vocab.itos[i] for i in takeuntil(i_eos, sent)])
                for sent in zip(*pred)]
126         all_pred_sents.extend(pred_sents)
127 
128         if phase == Phase.Test:
129             z_iter = zip(article_ids, gold_sents, pred_sents, latest_vals,
                    latest_closing_vals, attn_weights, input_data)
130             for (article_id, gold_sent, pred_sent, latest_val, latest_closing_val,
                    att_we, input_d) in z_iter:
131 
132                 bleu = sentence_bleu([gold_sent],
133                                      pred_sent,
134                                      smoothing_function=SmoothingFunction().method1
                                          )
135 
136                 gold_sent_num = replace_tags_with_vals(gold_sent,
                        latest_closing_val, latest_val)
137                 all_gold_sents_with_number.append(gold_sent_num)
138 
139                 pred_sent_num = replace_tags_with_vals(pred_sent,
                        latest_closing_val, latest_val)
140                 all_pred_sents_with_number.append(pred_sent_num)
141 
142                 description = \
143                     '\n'.join(['=== {} ==='.format(phase.value.upper()),
144                                'Article ID: {}'.format(article_id),
145                                'Gold (tag): {}'.format(', '.join(gold_sent)),
146                                'Gold (num): {}'.format(', '.join(gold_sent_num)),
147                                'Pred (tag): {}'.format(', '.join(pred_sent)),
148                                'Pred (num): {}'.format(', '.join(pred_sent_num)),
149                                'BLEU: {:.5f}'.format(bleu),
150                                'Loss: {:.5f}'.format(loss.item() / max_n_tokens),
151                                'Latest: {:.2f}'.format(latest_val),
152                                'Closing: {:.2f}'.format(latest_closing_val)])
153                 showAttention(input_d, pred_sent, att_we, dest_dir / Path(
                        article_id + ".png"))
154                 logger.info(description)  # TODO: info → debug in release
155 
156         accum_loss += loss.item() / max_n_tokens
157 
158     return RunResult(accum_loss,
159                      all_article_ids,
160                      all_gold_sents,
161                      all_gold_sents_with_number,
162                      all_pred_sents,
163                      all_pred_sents_with_number)
```

### B.3.2 ネットワーク構成

Listing B.8: Transcribe

```
1 import itertools
2 from collections import OrderedDict
3 from typing import List, Tuple, Union
```

```
4
5   import torch
6   from torch import Tensor, nn
7   from torchtext.data import Batch
8
9   from util.config import Config
10  from util.constant import (
11      GENERATION_LIMIT,
12      N_LONG_TERM,
13      N_SHORT_TERM,
14      TIMESLOT_SIZE,
15      Phase,
16      SeqType
17  )
18  from util.conversion import stringify_ric_seqtype
19
20
21  class Attention(nn.Module):
22      '''This implementation is based on `Luong et al. (2015) <https://arxiv.org/abs
            /1508.04025>`_.
23      '''
24
25      def __init__(self):
26          super(Attention, self).__init__()
27
28      def forward(self, h_t: Tensor, h_s: Tensor) -> Tensor:
29          return self.align(h_t, h_s)
30
31      def align(self, h_t: Tensor, h_s: Tensor) -> Tensor:
32          r'''
33          .. math:
34              a_{ij} =
35                  \frac{%
36                      \exp\left(
37                          \operatorname{score}\left(
38                              \boldsymbol{h}^\text{target}_j, \boldsymbol{h}^\text{
                                    source}_i
39                          \right)
40                      \right)
41                  }{%
42                      \sum_{\iota = 1}^I
43                          \exp\left(
44                              \operatorname{score}\left(
45                                  \boldsymbol{h}^\text{target}_j, \boldsymbol{h}^\
                                        text{source}_\iota
46                              \right)
47                          \right)
48                  }
49          '''
50          return nn.functional.softmax(self.score(h_t, h_s), dim=1)
51
52      def score(self, h_t: Tensor, h_s: Tensor) -> Tensor:
53          raise NotImplementedError
54
55
56  class GeneralAttention(Attention):
57
58      def __init__(self, h_t_size: int, h_s_size: int):
59          super(Attention, self).__init__()
60          r'''
61          Args:
62              h_t_size (int): the size of target hidden state
63              h_s_size (int): the size of source hidden state
```

```
64
65          This calculates scores by
66          ..math:
67              \boldsymbol{h}^{target}_j
68                  \cdot
69              \boldsymbol{W}^\text{attn} \boldsymbol{h}^\text{source}_i.
70          '''
71          self.w_a = nn.Linear(h_s_size, h_t_size, bias=False)
72
73      def score(self, h_t: Tensor, h_s: Tensor) -> Tensor:
74          return torch.bmm(self.w_a(h_s), h_t.transpose(1, 2))
75
76
77  def setup_attention(config: Config, seqtypes: List[SeqType]) -> Union[None,
        Attention]:
78
79      # enc_time_hidden_size = config.time_embed_size * len(seqtypes)
80      enc_time_hidden_size = 138
81      if config.attn_type == 'general':
82          # h_t^^c2^^b7(W_a h_s)
83          return GeneralAttention(config.dec_hidden_size, enc_time_hidden_size)
84      else:
85          return None
86
87
88  class Encoder_nonHigh(nn.Module):
89      def __init__(self, config: Config, device: torch.device):
90
91          super(Encoder_nonHigh, self).__init__()
92          self.used_seqtypes = [SeqType.StdLong,
93                                SeqType.StdShort,
94                                SeqType.NormMovRefLong,
95                                SeqType.NormMovRefShort] \
96              if config.use_standardization \
97              else [SeqType.NormMovRefLong,
98                    SeqType.NormMovRefShort]
99          self.used_rics = config.rics
100         self.use_extra_rics = len(self.used_rics) > 1
101         # self.base_ric = config.base_ric
102         # self.extra_rics = [ric for ric in self.used_rics if ric != self.base_ric]
103         self.extra_rics = [ric for ric in self.used_rics]
104         self.base_ric_hidden_size = config.base_ric_hidden_size
105         self.ric_hidden_size = config.ric_hidden_size
106         self.hidden_size = config.enc_hidden_size
107         self.n_layers = config.enc_n_layers
108         # self.prior_encoding = int(self.base_ric in self.used_rics)
109         self.dropout = config.use_dropout
110         self.device = device
111         self.attn_vec_type = config.attn_vec_type
112
113         self.use_dropout = config.use_dropout
114         #self.ric_seqtype_to_mlp = dict()
115
116         for (ric, seqtype) in itertools.product(self.used_rics, self.used_seqtypes)
              :
117             input_size = N_LONG_TERM \
118                 if seqtype.value.endswith('long') \
119                 else N_SHORT_TERM
120             """
121             output_size = self.base_ric_hidden_size \
122                 if ric == self.base_ric \
123                 else self.ric_hidden_size
124             """
```

```
125             output_size = input_size
126             """
127             mlp = MLP(input_size,
128                       self.hidden_size,
129                       output_size,
130                       n_layers=self.n_layers).to(self.device)
131             self.ric_seqtype_to_mlp[(ric, seqtype)] = mlp
132             """
133
134         lengths = [N_LONG_TERM if seqtype.value.endswith('long') else N_SHORT_TERM
135                    for (_, seqtype) in itertools.product(self.used_rics, self.
                       used_seqtypes)]
136         self.total_length = sum(lengths)
137         self.cat_hidden_size = \
138             self.total_length if len(self.used_rics) == 1 \
139             else 1 # self.prior_encoding * len(self.used_seqtypes) * self.
                    base_ric_hidden_size + \
140             #(len(lengths) - self.prior_encoding * len(self.used_seqtypes)) * self.
                    ric_hidden_size
141         """
142         self.cat_hidden_size = \
143             total_length + self.prior_encoding * len(self.used_seqtypes) * self.
                    base_ric_hidden_size \
144             if len(self.used_rics) == 1 \
145             else self.prior_encoding * len(self.used_seqtypes) * self.
                    base_ric_hidden_size + \
146             (len(lengths) - self.prior_encoding * len(self.used_seqtypes)) * self.
                    ric_hidden_size
147         """
148         self.dense = nn.Linear(self.cat_hidden_size, self.hidden_size)
149         if self.attn_vec_type == "concat":
150             self.attn_linear = nn.Linear(1, self.hidden_size)
151         elif self.attn_vec_type == "cnn":
152             self.window = 3
153             self.move = 1
154             self.attn_linear = nn.Linear(self.window, self.hidden_size)
155
156         if self.use_dropout:
157             self.drop = nn.Dropout(p=0.30)
158
159     def forward(self,
160                 batch: Batch,
161                 mini_batch_size: int) -> Tuple[Tensor, Tensor]:
162
163         L = OrderedDict()  # low-level representation
164
165         attn_vector = []
166
167         for (ric, seqtype) in itertools.product(self.used_rics, self.used_seqtypes)
                :
168
169             vals = getattr(batch, stringify_ric_seqtype(ric, seqtype)).to(self.
                    device)
170
171             if seqtype in [SeqType.NormMovRefLong, SeqType.NormMovRefShort]:
172                 # Switch the source to one which is not normalized
173                 # to make our implementation compatible with Murakami 2017
174                 L_seqtype = SeqType.MovRefLong \
175                     if seqtype == SeqType.NormMovRefLong \
176                     else SeqType.MovRefShort
177                 L[(ric, seqtype)] = getattr(batch, stringify_ric_seqtype(ric,
                        L_seqtype)).to(self.device)
178             else:
```

```
                L[(ric, seqtype)] = vals
        """
        for ric in self.used_rics:
            attn_vector.extend([num for num in [L[(ric, seq)] for seq in self.
                used_seqtypes]])
        """
        if self.attn_vec_type == "concat":
            attn_vector = torch.cat(list(L.values()), 1)
        elif self.attn_vec_type == "cnn":
            head = 0
            for ric in self.used_rics:
                seq_vec_list = [L[(ric, seq)] for seq in self.used_seqtypes]
                while head + self.window <= len(seq_vec_list):
                    attn_vector.extend([vec for vec in seq_vec_list[head:head+self.
                        window]])
                    self.head += self.move
                head = 0

        org_enc_hidden = torch.cat(list(L.values()), 1)  # Murakami model

        enc_hidden = self.dense(org_enc_hidden)

        if self.use_dropout:
            enc_hidden = self.drop(enc_hidden)
            org_enc_hidden = self.drop(org_enc_hidden)
            attn_vector = self.drop(attn_vector)

        if len(attn_vector) > 0:
            #attn_vector = torch.cat(attn_vector, 1)
            if self.attn_vec_type == "concat":
                attn_vector = attn_vector.view(mini_batch_size, self.total_length,
                    1)
            elif self.attn_vec_type == "cnn":
                org_enc_hidden = attn_vector
                attn_vector = torch.cat(attn_vector, 1)
                attn_vector = attn_vector.view(mini_batch_size, -1, self.window)
            attn_vector = self.attn_linear(attn_vector)

        return (org_enc_hidden, enc_hidden, attn_vector)



class AttnDecoder(nn.Module):
    def __init__(self,
                 config: Config,
                 output_vocab_size: int,
                 attn: Union[None, Attention],
                 device: torch.device):

        super(AttnDecoder, self).__init__()

        self.device = device
        self.dec_hidden_size = config.dec_hidden_size
        self.word_embed_size = config.word_embed_size
        self.time_embed_size = config.time_embed_size
        self.attn = attn

        self.word_embed_layer = nn.Embedding(output_vocab_size, self.
            word_embed_size, padding_idx=0)
        self.time_embed_layer = nn.Embedding(TIMESLOT_SIZE, self.time_embed_size)
        self.output_layer = nn.Linear(self.dec_hidden_size, output_vocab_size)
        self.softmax = nn.LogSoftmax(dim=1)

```

```
238         self.dec_hidden_size = self.dec_hidden_size
239         self.input_hidden_size = self.time_embed_size + self.word_embed_size
240         self.recurrent_layer = nn.LSTMCell(self.input_hidden_size, self.
                dec_hidden_size)
241 
242         self.enc_hidden_size = config.enc_hidden_size
243         attn_size = self.enc_hidden_size + self.dec_hidden_size
244         self.linear_attn = nn.Linear(attn_size, self.dec_hidden_size)
245 
246     def init_hidden(self, batch_size: int) -> Tuple[Tensor, Tensor]:
247         zeros = torch.zeros(batch_size, self.dec_hidden_size, device=self.device)
248         self.h_n = zeros
249         self.c_n = zeros
250         return (self.h_n, self.c_n)
251 
252     def forward(self,
253                 word: Tensor,
254                 time: Tensor,
255                 seq_ric_tensor: Tensor,
256                 batch_size: int) -> Tuple[Tensor, Tensor]:
257 
258         weight = 0.0
259         word_embed = self.word_embed_layer(word).view(batch_size, self.
                word_embed_size)
260         time_embed = self.time_embed_layer(time).view(batch_size, self.
                time_embed_size)
261         stream = torch.cat((word_embed, time_embed), 1)
262 
263         self.h_n, self.c_n = self.recurrent_layer(stream, (self.h_n, self.c_n))
264         output, hidden = self.c_n, self.h_n
265 
266         if isinstance(self.attn, Attention):
267             _, num_copy, _ = seq_ric_tensor.size()
268             copied_hidden = hidden.unsqueeze(1)
269             weight = self.attn(copied_hidden, seq_ric_tensor)
270             weighted_ric = torch.bmm(weight.view(batch_size, -1, num_copy),
271                                      seq_ric_tensor.view(batch_size, num_copy, -1))
272             weighted_ric = weighted_ric.squeeze()
273             hidden = torch.tanh(self.linear_attn(torch.cat((hidden, weighted_ric),
                    1)))
274             self.h_n = hidden
275 
276         output = self.softmax(self.output_layer(hidden))
277         return (output, weight)
278 
279 
280 class MLP(nn.Module):
281     def __init__(self,
282                  input_size: int,
283                  mid_size: int,
284                  output_size: int,
285                  n_layers: int = 3,
286                  activation_function: str = 'tanh'):
287         '''Multi-Layer Perceptron
288         '''
289 
290         super(MLP, self).__init__()
291         self.n_layers = n_layers
292 
293         assert(n_layers >= 1)
294 
295         if activation_function == 'tanh':
296             self.activation_function = nn.Tanh()
```

```
297         elif activation_function == 'relu':
298             self.activation_function = nn.ReLU()
299         else:
300             raise NotImplementedError
301 
302         self.MLP = nn.ModuleList()
303         if n_layers == 1:
304             self.MLP.append(nn.Linear(input_size, output_size))
305         else:
306             self.MLP.append(nn.Linear(input_size, mid_size))
307             for _ in range(n_layers - 2):
308                 self.MLP.append(nn.Linear(mid_size, mid_size))
309             self.MLP.append(nn.Linear(mid_size, output_size))
310 
311     def forward(self, x: Tensor) -> Tensor:
312         out = x
313         for i in range(self.n_layers):
314             out = self.MLP[i](out)
315             out = self.activation_function(out)
316         return out
317 
318 
319 class EncoderDecoder(nn.Module):
320     def __init__(self,
321                  encoder: Encoder_nonHigh,
322                  decoder: AttnDecoder,
323                  device: torch.device):
324         super(EncoderDecoder, self).__init__()
325 
326         self.device = device
327         self.encoder = encoder.to(self.device)
328         self.decoder = decoder.to(self.device)
329 
330         self.weight_lambda = 10 ** 0  # for supervised attention
331 
332     def forward(self,
333                 batch: Batch,
334                 mini_batch_size: int,
335                 tokens: Tensor,
336                 time_embedding: Tensor,
337                 criterion: nn.NLLLoss,
338                 phase: Phase) -> Tuple[nn.NLLLoss, Tensor, Tensor]:
339 
340         self.decoder.init_hidden(mini_batch_size)
341         concated_input, self.decoder.h_n, attn_vector = self.encoder(batch,
                mini_batch_size)
342 
343         loss = 0.0
344         n_tokens, _ = tokens.size()
345         decoder_input = tokens[0]
346         time_embedding = time_embedding.squeeze()
347 
348         pred = []
349         attn_weight = []
350         pred.append(decoder_input.cpu().numpy())
351 
352         if phase == Phase.Train:
353             for i in range(1, n_tokens):
354                 decoder_output, weight = \
355                     self.decoder(decoder_input, time_embedding, attn_vector,
                            mini_batch_size)
356                 loss += criterion(decoder_output, tokens[i])
357 
```

```
358                 topv, topi = decoder_output.data.topk(1)
359                 pred.append([t[0] for t in topi.cpu().numpy()])
360                 if self.decoder.attn:
361                     weight = weight.squeeze()
362                     attn_weight.append(weight)
363 
364                 decoder_input = tokens[i]
365 
366         else:
367             for i in range(1, GENERATION_LIMIT):
368                 decoder_output, weight = \
369                     self.decoder(decoder_input, time_embedding, attn_vector,
                            mini_batch_size)
370                 if i < n_tokens:
371                     loss += criterion(decoder_output, tokens[i])
372 
373                 topv, topi = decoder_output.detach().topk(1)
374                 pred.append([t[0] for t in topi.cpu().numpy()])
375                 if self.decoder.attn:
376                     weight = weight.squeeze(2).cpu().detach().numpy()
377                     attn_weight.append(weight)
378 
379                 decoder_input = topi.squeeze()
380         if phase == Phase.Test:
381             return (loss, pred, attn_weight, concated_input)
382         else:
383             return (loss, pred, attn_weight)
```