修　　士　　論　　文

発話状況を考慮したニューラル雑談対話モデル

Situation-Aware Neural Conversation Model

指導教員　　**喜連川 優教授**

東京大学 大学院情報理工学系研究科
電子情報学専攻

氏 名　　**48-156419 佐藤 翔悦**

提 出 日　　**平成29年2月3日**

概要

計算機による対話システムは，自然言語を用いた情報機器へのアクセス手段，コールセンターなどにおける人間のオペレータの代替手段としての期待から研究が行われてきた．こうした多くの対話システムは情報検索をはじめとする何らかの目的を達成するためのアプリケーションであり，タスク指向型対話システムと呼称される．

しかし，タスク指向型対話システムに対してもユーザは一定のコミュニケーション能力を期待することに加えて，対話システムの役割が高度化するにつれて過去の会話や現在の時間，ユーザの性格といったユーザの発言の背後にある様々な発話状況を考慮する事が不可欠になる．そのため，雑談対話において不可欠な「空気を読む」能力はタスク指向型対話システムにおいても非常に重要な能力となる．

本研究では会話データから陽に取得可能な発話状況に加え，発話内容のクラスタリングに基づく分類によって獲得した発話状況を対話モデルに導入することによって応答性能の向上を図る．またそうして得られた発話状況をより効果的に活用すべく，大域的なモデルに加え発話状況を考慮した局所的なモデルを併用することで状況に応じた応答が可能なニューラル雑談対話モデルを提案する．

実験ではマイクロブログから取得した会話データを用いた教師あり学習によってシステムを構築し，発話に対する応答候補からの選択タスクによって評価を行った．その結果，提案手法が平均的に最も良い結果を示すことを確認した．

謝辞

この 2 年間，研究室の皆様には大変お世話になりました．まず，指導教員の喜連川優教授に感謝致します．研究を行う上でこれ以上無い環境の中，楽しく研究を続けさせて頂きました．先生が仰った「本当に解くべき重要な問題とは何かについて日々苦しみながら考える」ということを忘れずこれからも研究を続けて行きたいと思っています．

次に，豊田正史准教授には学会の運営や授業などで多忙の中，多くのアドバイスを頂きました．また，多大な労力を掛けて収集して頂いている会話データは本研究を進める上で大きな助けとなりました．

吉永直樹准教授には 1 年次の頃から技術的な質問に始まり論文の添削，研究の方針など本当に数多くの事をご指導頂きました．いつも締め切り直前まで論文の添削にお付き合い頂き本当に申し訳ありません．

続いて，伊藤正彦特任准教授と横山大作特任助教に感謝致します．毎週の修士ミーティングや輪講の発表練習などにおいて，自分の拙い発表に対して辛抱強く意見をして頂き本当にありがとうございます．

小宮山純平助教と梅本和俊特任助教にも研究を進める上でいろいろなアドバイスを頂きました．自分の専門外の分野における知識や異なる視点からの意見は大変貴重でした．

先輩の石渡祥之佑さんとは研究室で話すことが多く，自分のアイディアや既存の手法に関する意見交換，論文についての相談など多くの事で助けられました．博士過程でもよろしくお願いします．

同期の岩成達哉くん，小泉実加さんとはお互い助け合ったというよりはどちらかというと助けられてばかりだった気もします．大学や学会の手続き・書類など，万

事適当な自分は几帳面な岩成くんに助けられることが本当に多かったです．また，論文やスライドの図を作るのが非常に苦手な自分にとって小泉さんの助力はとても大きなものでした．締め切り間際に手伝ってくれた時は本当にありがとう．

ここで名前を挙げた方々に留まらず，研究室の先生や先輩，後輩の皆さん，秘書の方々にも研究への意見に留まらず，日常生活の様々な面で大変お世話になりました．特に，後輩の赤崎智くんは研究についての議論にはじまり単位関係などで大きな助けとなってくれました．どうか無事卒業出来ることを祈って下さい．

最後に学費や生活費をはじめとして，学生生活を続ける上で大きな助けとなってくれている両親に感謝の意を捧げます．大学院はなかなか楽しいところでした．

2017 年 2 月 3 日

目 次

図目次

表目次

第1章　はじめに

1.1　背景

計算機による対話システムは，自然言語を用いた情報機器へのアクセス手段，コールセンターなどにおける人間のオペレータの代替手段としての期待から研究が行われてきた．特に近年では，Apple の “Siri” や NTT ドコモの “しゃべってコンシェル” などといったモバイル機器の操作のための知的エージェントが広く普及しており，より洗練された対話システムの重要性が高まっている [1].

こうした対話システムは 2 種類に大別される．質問応答や自動案内システムなどをはじめとした，特定の目的を達成するための対話システムをタスク指向型対話システムと呼び，一方で子供や老人の話し相手としての対話システムをはじめとした，コミュニケーションそのものを目的とする対話システムを非タスク指向型対話システムと呼ぶ.

しかし例えば “Siri” が本来，Web サービスや情報端末内のアプリケーションの検索をはじめとする秘書機能のために設計されたタスク指向型対話システムであるにも関わらず，「Apple をどう思いますか？」などといった質問によってコミュニケーションを取ろうとするユーザが存在する．このように，タスク指向型対話システムに対してもユーザは一定のコミュニケーション能力を期待するため，知的エージェントの持つコミュニケーション能力がもたらす快適さや生産性の向上についても注目されている [2].

また知的エージェントに求める役割が大きくなるにつれ，タスクの達成のためには以前の会話内容やユーザの意図や好み，時節や場所，今の話題といった様々な発話状況（ドメインと呼ぶ）を考慮に入れ，雑談を行う上で不可欠な「空気を読む」能

力が必要となる．そのため，非タスク指向型対話における雑談能力の追求はタスク指向型対話システムのみならず，言語を用いたコミュニケーションを必要とする多くのシステムにおいて非常に重要である．しかしながら従来の実用的な対話システムは，ルールベース型対話システムと呼ばれる特定の用途・状況を想定してパターン等を作りこむ事で実現されているシステムである．そのため，雑談において無数に存在する発話状況に対応した会話が可能なエージェントの構築は，そのコストの問題から現実的には非常に困難である．

そのため近年では統計的対話モデルと呼ばれる，Twitter などのマイクロブログから取得可能な幅広い話題を含む大規模な会話データを用いて，教師あり学習に基づき人手のパターンによらない雑談対話を実現しようという研究 [3, 4] や，統計的対話モデルと従来のルールベース対話モデルとの融合に関する研究 [5] が盛んに行われている．しかしこれらの研究においても主に学習データ中の発話・応答のみを考慮する事が多く，陽に発話状況を考慮したものは我々の知る限り存在しない．

一方で学習データ全体から着目した発話状況における会話を抽出することで発話状況を考慮し，高品質な対話システムを構築する研究 [6] が存在するが，膨大なデータから適切な発話状況を切り出す難しさから，十分にその有効性が検討されているとは言いがたい．またこうしたある発話状況における会話データのみを用いて学習を行うというアプローチは，その応答がより発話状況を捉えたものになる一方で学習データの減少に繋がるため，入手した会話データ全てを有効に活用しきれていないという問題点も存在する．

1.2 本研究の提案

本研究では発話内容に基づく話題・文脈といった情報を内的な発話状況，発話時間・場所や発話者間の人間関係などに基づく情報を外的な発話状況と位置づけて，発話内容から抽出した内的な発話状況と，自明に取得可能な外的な発話状況を用いることによる応答の改善を試みる．

これに加え，ある発話状況下におけるデータから学習を行う局所的なモデルと学

習データ全体から学習を行う大域的なモデルを並列に用いる事で発話状況への特化と学習データサイズとの間のトレードオフを解決するニューラル対話モデルを提案する．入力としては発話・応答を構成する単語列の組に加え，その発話・応答がおかれた発話状況を表す ID が与えられた上で，発話に対応する応答の確率を最大化するように学習が行われる．

前者については第 4 章，後者については第 5 章においてその詳細を述べる．評価については，ある発話に対する応答候補からの選択タスクによって既存手法との比較を行う．評価の際は入力として発話と複数の応答候補が与えられ，発話とそれぞれの応答の組に対し，モデルが推測する応答の確率によって応答候補が順位付けされる．

1.3 論文の構成

以降，本論文は以下のように構成されている．

第 2 章 本研究の実験において用いた対話モデルに関する基礎知識，特に中心的な技術となるニューラルネットワークについて述べる．

第 3 章 対話応答，またそれ以外の自然言語処理タスクにおけるドメイン適応についての関連研究，また近年の教師あり学習に基づく対話モデルにおける成果について述べる．

第 4 章 クラスタリングに基づき発話内容から内的な発話状況の分類を行い，発話状況ごとに独立にモデルを訓練することでそれぞれの発話状況を考慮した対話モデルを得た上で，応答選択タスクによって評価する．また，NTCIR-12 Short Text Conversation Japanese タスクにおける我々のシステムと他チームの結果についても比較する．

第 5 章　ある発話状況下におけるデータから学習を行う局所的なモデルと学習データ全体から学習を行う大域的なモデルを並列に用いる事で，発話状況への特化と学習データサイズとの間のトレードオフを解決する local/global SEQ2SEQ を提案する．また，提案モデルに対していくつかの内的・外的な発話状況を用いて，その応答性能への影響を応答選択タスクによって評価する．

第 6 章　本研究のまとめを行う．

第 7 章　本研究における実験・分析をもとに，今後の課題と展望について述べる．

第2章　基礎知識

本節では，本研究のために構築した対話システムの中心となる技術であるニューラルネットワークについて自然言語処理への利用という側面から述べる．機械学習においてニューラルネットワークと呼ばれるモデルはいくつかの種類が存在するが，本節では本研究に関連するものとして，順伝播型ニューラルネットワーク (Feed-forward Neural Network) の基礎的な仕組みについて述べ，系列データに対して拡張を行った再帰型ニューラルネットワーク (Recurrent Neural Network) やその派生モデルとしての Sequence-to-Sequence(SEQ2SEQ) についても説明を行った後，ニューラルネットワークを用いて獲得される語の意味の数学的表現についても述べる．

近年，画像処理におけるニューラルネットワークの成功 [7] をきっかけとしてその有用性への注目が分野を超えて高まり，それに伴い機械翻訳 [8] や文書要約 [9]，対話応答 [10] など，多くの自然言語処理タスクにおいてその効果が報告されている．

その理由としてはそのモデルの構造から，目的関数の最適化に伴い入力を表す特徴量が人手によらず獲得される，という点が大きい．また，画像処理における画素や自然言語における単語・文字のように，入出力として取りうる要素やその組み合わせの種類が非常に多い場合，単純にその要素を特徴量とした場合，学習データ中に存在しない組み合わせに対する性能の低下が懸念される．こうした問題はデータスパースネス問題と呼ばれる．しかし，ニューラルネットワークにおいてはその高い汎化性能によって，この問題に対して頑健であることも大きな利点である．

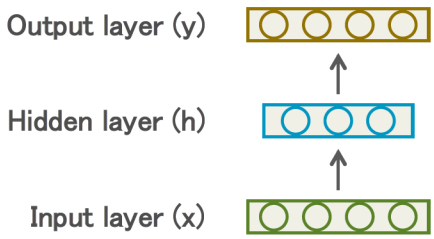

図 2.1: Feed-forward Neural Network

2.1 入出力

多層パーセプトロンと呼ばれる分類問題における一般的なニューラルネットワーク（ここでは簡単のため，隠れ層は 1 層とする）は入力層 (input layer) $\mathbf{x} = (x_1, \dots, x_n)^{\mathrm{T}}$ と隠れ層 (hidden layer) $\mathbf{h} = (h_1, \dots, h_d)^{\mathrm{T}}$，出力層 (output layer)$\mathbf{y} = (y_1, \dots, y_m)^{\mathrm{T}}$ からなる（図 2.1）．n は入力となる特徴量の次元，m は分類先のクラス数であり，d は対象とするタスクの複雑さに応じてハイパーパラメータとして与えられる．

隠れ層 $\mathbf{h}$，出力層 $\mathbf{y}$ は以下の式によって計算される．

$$\begin{aligned} \mathbf{h} &= f_h\left(\mathbf{W}_x\,\mathbf{x} + \mathbf{b}_h\right) \\ \mathbf{y} &= f_y\left(\mathbf{W}_y\,\mathbf{h} + \mathbf{b}_y\right) \end{aligned} \tag{2.1}$$

隠れ層は入力を抽象化した内部表現に相応する．隠れ層は 1 層であることをを前提としたが，多層に設定する場合隠れ層 $\mathbf{h}^l$ の値を用いて後段の隠れ層 $\mathbf{h}^{l+1}$ を計算することになる．一般に層を増やすほど抽象化された入力の潜在情報が階層的に獲得可能であるため，十分に例が存在する場合タスクの精度が向上するとされる．その上で隠れ層の情報を元に，それぞれのクラスに対する確率が出力層として得られる．

$\mathbf{W}_h, \mathbf{W}_y$ は重み行列，$\mathbf{b}_h, \mathbf{b}_y$ はバイアス項と呼ばれ，それぞれの層が持つパラメータである．これらをまとめて $\boldsymbol{\theta}$ とする．

$$\boldsymbol{\theta} = (\mathbf{W}_h, \mathbf{W}_y, \mathbf{b}_h, \mathbf{b}_y) \tag{2.2}$$

多層パーセプトロンの学習においては第 2.2 節で述べる手法を用いて学習データからパラメータ $\boldsymbol{\theta}$ の学習を行うことが目的となる．

また f_h, f_y は活性化関数と呼ばれ，各層の出力への非線形性の導入や正規化を目的として用いられる．隠れ層においては標準シグモイド関数

$$f(z) = \frac{1}{1+e^{-z}} \tag{2.3}$$

や双曲線正接関数

$$f(z) = tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \tag{2.4}$$

ReLU 関数

$$f(z) = max(0, z) \tag{2.5}$$

などの非線形関数がよく用いられる．一方，分類問題における出力層では出力としてそれぞれのクラスに対する確率を得ることが目的であるため，ソフトマックス関数

$$f(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{2.6}$$

を活性化関数として用いることによって出力の総和を 1 とする．

2.2 確率的勾配降下法

続いて，多層パーセプトロンのパラメータ $\boldsymbol{\theta}$ の学習について述べる．

分類問題を対象とする多層パーセプトロンは K 件の学習データが与えられた時，入力 $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_K)$ とそれらに対する出力 $\mathbf{Y} = (\mathbf{y}_1(\mathbf{x}_1; \boldsymbol{\theta}), \dots, \mathbf{y}_K(\mathbf{x}_K; \boldsymbol{\theta}))$，目標となる出力 $\mathbf{D} = (\mathbf{d_1}, \dots, \mathbf{d_K})$ を元に，誤差関数

$$E(\boldsymbol{\theta}) = -\sum_{i=1}^{K}\sum_{j=1}^{m} d_{ij} \log y_{ij}(\mathbf{x}_i; \boldsymbol{\theta})) \tag{2.7}$$

を最小化することでの学習を行う．$\mathbf{x}_i$ は入力特徴量を表す n 次元のベクトル，$\mathbf{y}_i, \mathbf{d}_i$ はそれぞれのクラスに対する確率を表す m 次元のベクトルであり，目標となる出力 $\mathbf{d}_i$ は

$$\mathbf{d}_i = (0, ..., 1, ..., 0)^{\mathrm{T}} \tag{2.8}$$

のように正解のクラスを表す次元のみが 1 であるベクトルとして与えられる．誤差関数 $E(\boldsymbol{\theta})$ は交差エントロピー (cross entropy) と呼ばれ，交差エントロピーを最小化することは出力の確率分布を最適化して目標の確率分布に近づけることに相応する．

ニューラルネットワークにおいてしばしば用いられる確率的勾配降下法 (stochastic gradient descent, SGD) と呼ばれる学習法では，学習データのあるサンプルに対する誤差関数を $E(\boldsymbol{\theta})$ とした時，以下の式によってパラメータ $\boldsymbol{\theta}$ の更新を行う．

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \epsilon \nabla E(\boldsymbol{\theta}^t) \tag{2.9}$$

一般的には学習データ全体から一部のデータを取り出し，それぞれのサンプルから得られた結果を用いてパラメータを更新する，という事を繰り返す．この方法はミニバッチ学習と呼ばれる．また，ϵ は学習係数 (learning rate) と呼ばれる一度の更新量の大きさを決めるハイパーパラメータであり，$\nabla E(\boldsymbol{\theta}^t)$ を勾配と呼ぶ．

2.3 誤差逆伝播法

第 2.2 節で述べた手法を用いる際，出力層のパラメータについては誤差関数から直接微分可能であるが隠れ層，特に多層の場合は入力に近い層になるほどその計算が困難になる．この問題を解決するため，誤差逆伝播法 [11] を用いる．

以下簡単のため，多層パーセプトロンにおける l 層目の隠れ層 (k 次元) を

$$\mathbf{h}^l = (1, h_1^l, ..., h_k^l)^{\mathrm{T}} \tag{2.10}$$

$l-1$ 層と l 層の間の重み行列の i 行目を

$$\mathbf{w}_i^l = (w_{i0}^l, ..., w_{ij}^l, ..., w_{ik}^l) \tag{2.11}$$

とすることで，隠れ層の常に 1 を取る次元に対する重みとしてバイアス項を統一的に扱う．この時

$$z_i^l = \sum_j w_{ij}^l h_j^{l-1} = \sum_j w_{ij}^l f(z_j^{l-1}) \tag{2.12}$$

とすると，ある重み w_{ij}^l の誤差関数への寄与は l 層の i 番目のユニットの出力のみであることから，勾配は

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial z_i^l}\frac{\partial z_i^l}{\partial w_{ij}^l} \tag{2.13}$$

と表せる．まず，第 2 項については式 (2.12) の定義から以下のように簡単に計算可能である．

$$\frac{\partial z_i^l}{\partial w_{ij}^l} = h_j^{l-1} \tag{2.14}$$

また第 1 項については，z_i^l の変動による誤差関数への寄与は $l+1$ 層目のそれぞれの出力を変化させることでのみ生じることから以下のように変形可能である．

$$\frac{\partial E}{\partial z_i^l} = \sum_k \frac{\partial E}{\partial z_k^{l+1}}\frac{\partial z_k^{l+1}}{\partial z_j^l} \tag{2.15}$$

ここで式 (2.15) の左辺を δ_i^l とおく．この時式 (2.14) から

$$\frac{\partial E}{\partial w_{ij}^l} = \delta_i^l h_j^{l-1} \tag{2.16}$$

である．式 (2.12) より，式 (2.15) 第 2 項が直接計算可能であることに注目すると式 (2.15) は

$$\delta_i^l = \sum_k \delta_k^{l+1}(w_{kj}^{l+1} f'(z_j^l)) \tag{2.17}$$

と変形可能である．出力層における δ_i^l は直接微分計算が可能であるため，式 (2.17) のもと出力層から入力層に向かって順に δ_i^l の計算を行うことで，式 (2.16) と合わせてすべての層における勾配が計算可能となる．

2.4 Recurrent Neural Network

ここまでで説明した多層パーセプトロンは順伝播型ニューラルネットワーク (Feed Forward Neural Network) と呼ばれるモデルであり，入力が固定長に限定されるため，音声やテキストをはじめとする可変長の系列データに対して適用することが困難であ

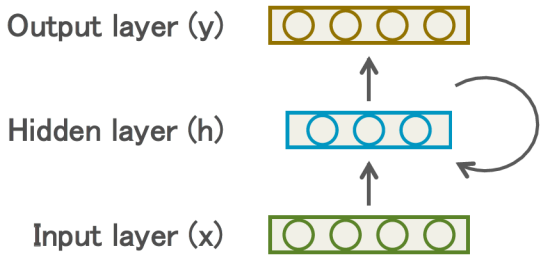

図 2.2: Recurrent Neural Network

るという問題点が存在する．そうしたデータに対しては，本節で述べる再帰型ニューラルネットワーク (Recurent Neural Network, RNN) を用いる．RNN (図 2.2) はモデル内に有向閉路を持つモデルであり，各時刻ごとに入力を受取り隠れ層の更新を行う事で系列データ全体の情報を考慮可能である．ある時刻 t の隠れ層 $\mathbf{h}(t)$，出力層 $\mathbf{y}(t)$ は以下の式によって計算される．

$$\begin{aligned} \mathbf{h}(t) &= f(\mathbf{W}_{hh}\mathbf{h}(t-1) + \mathbf{W}_{xh}\mathbf{x}(t) + \mathbf{b}_h) \\ \mathbf{y}(t) &= f(\mathbf{W}_{hy}\mathbf{h}(t) + \mathbf{b}_y) \end{aligned} \tag{2.18}$$

RNN の誤差逆伝播計算においても RNN を時間方向に展開した順伝播型ニューラルネットワークとして捉え，最終時刻から順に勾配計算を行うことで同様にパラメータの更新が可能である．

自然言語処理における最も一般的な用途としては，言語モデル [12, 13] と呼ばれる，ある言語における単語の並び順の自然さを表すモデルが挙げられる．RNN 言語モデルにおいては n 語の単語列に対し，i 番目までの単語が入力された時に $i+1$ 番目の単語を推測することが目標となる．このモデルに対する入力として，i 番目の単語 $\mathbf{x}(t)$ は以下のように与えられる．

$$\mathbf{x}(t) = (0, ..., 1, ..., 0) \tag{2.19}$$

式 (2.19) は 1-of-K 表現と呼ばれるベクトル表現であり，あらかじめ定められた入力単語を表す ID の次元の値が 1，それ以外の次元の値は 0 を取る．

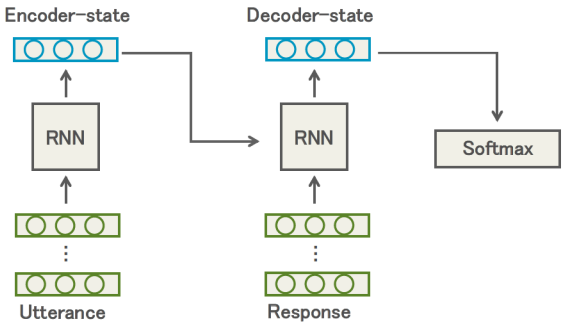

図 2.3: Sequence-to-Sequence (SEQ2SEQ)

多くの場合モデルの学習を行う際にハイパーパラメータとして語彙数 V が定められた上で，学習データにおける出現頻度順の上位 V 単語までを既知語とし，それ未満の単語はすべて未知語を表す ID に置き換えられる．語彙数 V を大きく設定するほど学習データ中の低頻度語についてより適切に解釈することが可能になる一方で，語彙数はモデルの計算コストに大きく影響するため，現実的には数万から数十万語程度に抑えざるを得ない．そうした問題を解決すべく，入力を単語ではなくその異なり数が小さい文字ベースで行おうとする試みも様々なタスクで存在し [14, 15]，計算コストの低さにも関わらず単語ベースのモデルと同程度，場合によっては上回る結果を残している．

また，RNN には Long-Short Term Memory (LSTM) [16, 17] や Gated Recurrent Unit(GRU) [18] をはじめとする複数の派生モデルが存在し，長期系列データに対する学習能力の高さからこれらの派生モデルが用いられることが多い．本研究の対話モデルにおいても，RNN として LSTM を採用した．

2.5 Sequence-to-Sequence

Sequence-to-Sequence (SEQ2SEQ) [19] モデルは Encoder-Decoder モデル [18] の一種であり，可変長の入力を受け取りそれに応じた可変長の出力を返すモデルである（図 2.3）．自然言語処理においては入出力は文であり，実際には文を構成する単語

列，もしくは文字列が系列データとして与えられる．

SEQ2SEQ モデルは Encoder, Decoder と呼ばれる 2 つの RNN からなる．まず Encoder で入力系列をすべて受け取った後，最終時刻の Encoder の出力を Decoder の初期状態とする．その上で Decoder 側で RNN と同様に各時刻の出力の計算を一定回数，もしくは出力終了を意味するクラスが選択されるまで行うことによって可変長の出力が可能である．モデルの学習はそれぞれの時刻におけるモデルの出力と目標となる出力に対し，クロスエントロピーの平均を最小化することで行われる．

これらのモデルは当初機械翻訳のモデルとして提案された [19] が，ある文を入力として受け取りそれに応じた文を出力として返す，というタスク間の類似点から対話応答においても多くの研究で採用されている [10][20].

2.6 語の意味の数学的表現

本研究を含む近年の多くの自然言語処理の研究は言語における単語や文を，その意味を表す実数値ベクトルとして表現した上で扱うという考えに基づいている．この語の意味表現としての実数値ベクトルは分散表現，または Embedding と呼ばれ，Hinton ら [21] によって提唱された．

分散表現を構築する手法については単語の共起行列に基づくものをはじめとして多くの手法が提案されてきたが，そのほとんどは分布仮説 [22] と呼ばれる，「ある語の意味はその周辺に存在する語の分布によって特徴づけられる」という考え方に基づく．中でも，Mikolov らの手法 [23, 24, 25] をはじめとした，ニューラルネットワークに基づく手法によって学習された分散表現は自然言語処理における様々なタスクで大きく性能向上に寄与している．

多くの場合，ニューラルネットワークにおける分散表現の学習は言語モデルの最適化によって行われる．以下では，RNN 言語モデルを例として分散表現の学習について述べる．

例えば，以下の文から言語モデルの学習を行う場合を考える．

I have a cute dog.

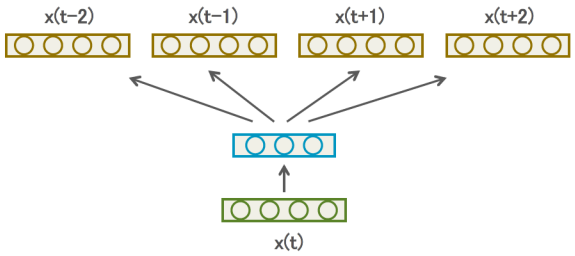

図 2.4: Skip-gram

ここで，第 2.4 節で述べたようにモデルの語彙数を V，隠れ層の次元を d とすると，ある入力単語の 1-of-K 表現は，$\mathbf{W}_{xh} \in \mathbb{R}^{d \times V}$ によって実数ベクトルに変換された上で隠れ層に入力される．言語モデルは単語の並び，つまり単語の共起情報を元に次の単語を推定するモデルであるため，“cute” という単語を入力することで “dog” をはじめとする何かしらの “cute” なものが周囲に共起するということが学習される．言い換えると，“cute” の 1-of-K 表現が $\mathbf{W}_{xh}$ によって変換された実数ベクトル $\mathbf{x}_{cute}$ はその共起情報を表すものとなり，分布仮説によるとそれは単語の意味であると考えられる．“cute” の分散表現とはこうして得られた d 次元のベクトル $\mathbf{x}_{cute}$ を指す．一般的には d の値としては 50〜1000 に設定される事が多い．

さらに “cute” に近い意味を持つ “pretty” という単語について考えると，$\mathbf{x}_{pretty}$ が入力されることによる周囲の単語の共起への影響は $\mathbf{x}_{cute}$ と近いものとなる．つまり類似する単語はその分散表現もベクトル空間上で近いものとなるため，コサイン類似度などの計算によって単語の意味の近さを数学的に計算可能である．また，分散表現はその意味の四則演算によって

$$\mathbf{x}_{King} - \mathbf{x}_{Man} + \mathbf{x}_{Woman} \simeq \mathbf{x}_{Queen}$$

のような，意味的な線形性を持つことが経験的に知られている [23].

より良い語の意味表現を構築しようとする研究 [26, 27] や同様の考え方に基いて文の分散表現を構築しようとする研究 [28, 29] は近年非常に盛んであり，いくつかの派生モデルが存在する．本研究で主に単語の分散表現として用いた用いた **word2vec** は，Mikolov らが提案した Skip-gram [25] と呼ばれるモデルによって構築された分

散表現を指す.

RNN 言語モデルが i 番目までの文脈単語から $i+1$ 番目の単語を推測するモデルであったのに対し，逆に Skip-gram (図 2.4) は文中のある単語に対する周囲の文脈単語の推測を行なうモデルであり，一般に RNN 言語モデルによって獲得された分散表現と比べて高性能なものが得られることが知られている.

第3章　関連研究

本章では関連研究について述べる．本稿では対話応答タスクにおいて，発話を行う際の時節・人間関係などを総称して発話状況と定義しているが，それ以外のタスクにおいては必ずしも対象とするデータが発話であるとは限らない．そのため本章では，データの背後に存在する様々なある種の一貫した特徴を総称してドメインと呼称する．

3.1　対話応答・機械翻訳における近年の研究

本節では対話応答・機械翻訳における関連研究について述べる．ここで機械翻訳に関する研究も紹介する理由としては，入出力が文，実際には文を構成する単語列や文字列であるという両タスクの共通点から，機械翻訳において有効な手法が対話応答でも適用可能である場合が多いためである [3, 30].

Higashinaka らは連続した発話の中で，それ以前の文脈を考慮した上で現在の対話行為（例: 挨拶，質問_事実，情報提供など）の種類を決定すべく，高頻度語の bag-of-words 特徴量を用いた無限 HMM によって発話クラスタリングを行っている [31]. 実験では，従来手法である Chinese Restaurant Process (CRP) に対して，系列的な情報を考慮することによってより高精度に発話の対話行為を認識することが出来ることを示した.

Ritter ら [3] はマイクロブログに含まれる膨大な対話データに着目し、発話から応答への翻訳と解釈することで統計的機械翻訳手法を用いて対話をモデル化し，統計的対話モデルの研究の道を開拓した。

Hasegawa らは発話が聞き手に対し喚起させる感情の種類に着目し，人手で作成し

た少数の規則によってTwitterから取得した大規模な対話データを怒り，喜び，悲しみなどといった9つのカテゴリに分類した感情タグ付き対話コーパスを構築している[6]．実験では，そのコーパスから統計的対話モデルを学習することで特定の感情を喚起するような応答の生成を試みている．

機械翻訳や文生成タスクにおける数々の研究[32, 33, 34]においてニューラルネットワークを用いたモデルの有効性が注目されるにつれ，対話応答タスクにおいてもニューラル翻訳モデルをベースとしたモデルが多くの研究において用いられている．続いて述べる2つのニューラルネットワークに基づく研究は，我々の研究にとって特に重要な位置を占める．

Liらは，統計的対話モデルにおける応答の一貫性の低さを問題とした[20]．統計的対話モデルは主に不特定多数の発話・応答者による会話データを用いてモデルの学習を行うことが多いため，「どこに住んでいますか？」と尋ねられた時は「東京です」と返答し，続いて「どこ住み？」と尋ねられた時は「大阪」と返答してしまうような例が頻発する．そのため，彼らは会話データをもとにユーザをいくつかのユーザタイプへとクラスタリングし，ニューラル対話モデルにおいてある応答があるユーザタイプによって行われた，という情報をuser-embeddingと呼ばれる実数値ベクトルとしてモデルに与えた上で応答の最適化を行うことで，ユーザタイプごとに一貫した応答を得ることに成功している．

Johnsonらは，ニューラル翻訳モデルを用いた多国語の機械翻訳において，翻訳先あるいは翻訳元の言語を表すembeddingを入力の先頭に加える，というシンプルな方法によって，複数の言語からなる学習データから単一のモデルの学習を行い，複数言語間での翻訳を可能にした[35]．

これらの2つの研究について，Liらの研究では発話者・応答者の性質，Johnsonらの研究では言語の差異を対象として，我々が対話応答タスクにおいて発話状況と呼ぶ，入出力文の背後に存在する一貫した特徴をモデルに取り入れることで性能を向上させている．両者の手法はそのネットワークの構造は異なるものの，発話状況をembeddingと呼ばれる数百次元の実数ベクトルとして表現した上でモデルに追加している，という点が共通する．

その一方で我々の本研究における提案手法では，学習データ全体の中に存在する発話状況の差異に対してモデルを独立に学習させることによって，それぞれのモデルはある発話状況における発話のより細かい差異を捉えた上で，そり適切に発話を解釈した応答を行うことが可能になる．

前者のアプローチと，後者のアプローチは異なる長所・短所を持ち，どちらが効果的であるかについては学習データの性質によって異なると考えられるため，定性的な議論は困難である．そのため，我々は第 5 章で行う実験において，彼らの手法を実装した上で提案手法との比較を行った．学習・評価データとしてはマイクロブログにおける会話データを用いることで，実際の対話応答に近い環境における比較となるように実験設定を行った．

3.2 ドメイン適応

第 1 章では，教師あり学習に基づく対話モデルにおいて，目的とする発話状況（ドメイン）において為された会話を学習データとして用いることによるシステムの性能の向上の可能性について述べた．

しかし現実には，対象とするドメインにおける学習データが十分に取得出来ない場合が存在する．そのため，ドメイン適応と呼ばれる，異なるドメインのコーパスから得られたモデルをタスクの対象となるドメインに適応させることを目的とした研究や，あるドメインにおけるデータの中でのサブドメインを形成することを目的とした研究が行われている．

本研究と基本的な考えを共通するものとして，機械翻訳タスクにおけるドメイン適応に関する Yamamoto らの研究がある [36]．彼らの手法では，各クラスタにおける言語モデルのパープレキシティを最小化するような形で文の組の再配置を繰り返すことで，詳細で一貫性のあるクラスタを得た後，それぞれのクラスタごとに翻訳モデルを構築している．実験では，日英翻訳タスクにおいて BLEU スコアの改善が得られることを確認した．しかし，異なるドメインに対してモデル単位で独立に学習を行うことによって，個々のモデルの学習データサイズが減少してしまうという

問題が残る．

Daume III ら [37] の研究，Kim らの研究 [38] は我々の研究と多くの共通点を持つ．Daume III らの提案した手法は Feature Augmentation と呼ばれ，学習データが 2 種類のドメイン s,t のどちらかに属し，かつ入力として特徴量ベクトル $\mathbf{x}$ が与えられた時に，$\mathbf{x_1} = \mathbf{x_2} = \mathbf{x}$ とした上でドメイン s では

$$\Phi^s = (\mathbf{x_1}, \mathbf{x_2}, \mathbf{0}) \tag{3.1}$$

と，ドメイン t では

$$\Phi^t = (\mathbf{x_1}, \mathbf{0}, \mathbf{x_2}) \tag{3.2}$$

とモデルへの入力を拡張する．こうすることによって，$\mathbf{x_1}$ に対応する部分についてはドメイン共通の要素を，$\mathbf{x_2}$ に対応する部分についてはドメイン固有の要素を学習することが可能になる．実験においては固有表現抽出をはじめとする様々なタスクによって，提案手法の有効性を確認した．

また Daume III らの研究に基づき，その手法をニューラルネットワークによるものに拡張したものが Kimra の研究 [38] である．彼らはドメインに対してそれぞれ学習を行った独立なネットワークと，ドメイン共通のネットワークによる出力を合わせる事でニューラルネットワークにおける Feature Augmentation を実現した．それに加えて，Shared Structure Learning Framework と呼ばれるドメイン共通部分の構造に若干の変更を加えたモデルの提案を行い，様々なドメインにおける系列ラベリングにおいてその性能の評価を行っている．

我々の研究とこれらの関連研究の大きな違いは，ドメインの差異を既知とするか否かにある．つまりこうした既存のドメイン適応の多くは主にコーパスの差異をドメインの違いとして扱っているのに対し，我々の研究は文書要約タスクにおける談話構造の利用 [39] や，機械翻訳タスクにおける句構造情報の利用 [40] に関する研究のように，陽に与えられてはいないがデータ中に存在すると考えられる差異の分類を行うことで，従来の学習モデルにおいて十分に意識することが出来ていなかった情報を取り入れ，応答性能の向上を行うものである．

第4章　発話内容から発話状況を推測する応答選択システム

提案手法では，雑談対話（データ）をより細かな発話状況（サブドメイン）に分割し，得られたサブドメインごとに複数の対話モデルを訓練することで，全データを用いた単一の対話モデルよりも良い性能が得られるのではないかという発想に基づいている．しかし当然ながらデータを分割すればするほど，一つのサブドメインあたりの学習データは少なくなる．従って，発話状況を考慮することによる性能の向上は学習データの減少による個々のモデルの性能の低下をどこまで担保出来るのかという点がこの研究において注目する点である．

図 4.1 に，我々の提案する対話システムの全体図を示す．以降で，順に各コンポーネントについて述べる．

4.1　発話内容の分散表現に基づく発話状況のクラスタリング

まず，我々は発話内容をクラスタリングすることで発話状況を得る．しかしオンライン対話では一つの発話が短いため，従来の bag-of-words 表現に基づき発話をクラスタリングするとデータの可塑性のため適切なクラスタリングが難しい．そこで，我々は word2vec と呼ばれる Mikolov ら [25] が提案した単語の分散表現（単語ベクトル）を用いて発話を密ベクトルで表現し，クラスタリングを行う．

クラスタリングの手順としてはまず，対話モデルの学習データとして，我々が対象とする Twitter から取得した雑談対話（発話-応答ペア）の各発話を形態素解析器

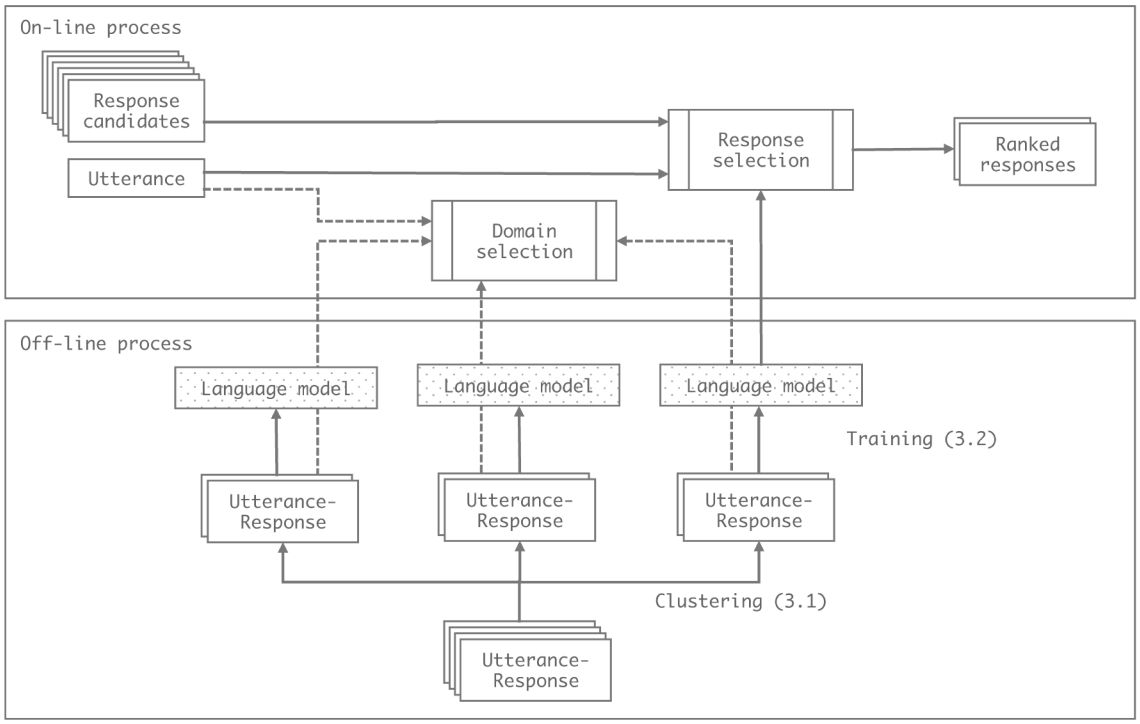

図 4.1: システムの全体図

MeCab[1] を用いて各形態素に分割する．この際，今回対象となるデータには新語が多く含まれていると予想されるため，それらに対応した辞書である mecab-ipadic-neologd[2] を用いた．このようにして得たそれぞれの発話中の形態素（単語）について，（事前に構築した）単語ベクトルの平均を取ることで発話全体のベクトル表現を構築する．その上で，発話のベクトル表現に対して k-means クラスタリングを適用し，発話状況の共通する発話-応答ペアを収集した．

[1] http://taku910.github.io/mecab/
[2] https://github.com/neologd/mecab-ipadic-neologd

4.2 Support Vector Machine を用いた応答候補の選択

第??節で述べる手法を用いた提案手法による応答の妥当性の判断は計算コストが大きく，数十万件ほどの応答候補から応答選択タスクを行う場合，それぞれの発話に対するその応答候補全ての妥当性を計算するのは現実的ではない．そのため，我々は二値分類器を用いて応答候補の事前絞込を行った．

具体的には、大量の訓練例からの非線形学習をサポートした kernel slicing [41] を実装した Opal[3] を用いて，発話と応答の bag-of-words (BOW) を素性とする分類器を学習した．この際に，カーネルとして二次の多項式カーネルを利用することで、発話と応答の BOW の組み合わせを考慮することが可能になり、発話に対する適切な応答に含まれる単語の組み合わせを捉えている．

学習データとしては，各発話に対して実際の応答との組み合わせを正例とし，応答候補からランダムに選んだ応答を正例と同じ数だけ負例として与えている．その後，テストデータの発話に対してそれぞれの応答候補の二値分類を行い，その分離平面からのマージンの大きさによって順位付けしている．

4.3 Recurrent Neural Network 言語モデルを用いた応答候補の順位付け

第??節で述べた手法に基いて対話データから得られたサブドメインごとの対話データを学習データとして，第 2.4 節で述べた RNN 言語モデルを用いて応答選択を行う．

本節で行う実験では，この RNN 言語モデルを応用して応答選択モデルを構築している．具体的には，会話データにおけるある発話とそれに対する応答を，区切りとなるトークンを挟んで繋げた一文を 1 つの発話・応答の組として与え，応答を構成する単語列のソフトマックス損失の平均によって応答候補の順位付けを行った．ソフトマックス損失は 2 つの異なる確率分布の間に定義される尺度であり，値が低いほど適切な応答とみなすことができる．

[3] http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/opal/

応答選択の手順としては，まずテストデータに含まれる各発話に対して，第 4.1 節と同様にベクトル表現を与える．次に，第 4.1 節で得られたサブドメインとなるクラスタのうち，中心がこの発話ベクトルから最も近いものを選択し，発話状況を推定する．最後に，選ばれたサブドメインのクラスタから作られた言語モデルを応答選択に用いる．今回我々の応答モデルは長期依存が存在する時系列データに対して有効な LSTM[16] を用いた．

この際，システムは 2 種類の方法で発話に対する応答が可能である．1 つが既に存在する応答候補を与えられた発話と組にして与えることで応答候補からの選択を行うというものであり，もう 1 つが発話と区切りトークンのみを与え，その入力を前提とした上で最尤の単語を選択し，入力に追加する．これを文の終端記号（EOS）が登場するまで繰り返して応答を生成する，というものである．後者は多くの応答候補に対するスコアを計算する必要がない点において低コストであるが，生成された応答がどの程度適切であるかを客観的な基準に基いて自動評価するのは困難である．そのため，本節では前者の応答候補からの選択によって評価を行う．

4.4 実験

本章において，我々は主に以下の 2 つの実験を行った．

1. 人手によらない低コストな評価方法として，小規模な応答候補から選択された結果が正解の応答を含む割合による評価実験を行った．応答選択には第 4.1 節，第 4.3 節の手法を用いた．

2. NTCIR-12 Short Text Conversation タスク [42] においてシステム全体の評価のために行った実験であり，膨大な応答候補の中から第 4.1 節，第 4.2 節，第 4.3 節の全ての手法を使って選択された応答の人手評価を行った．

また予備実験として，発話・応答の間では近い意味の単語が出現しやすいという経験則から，発話・応答に含まれるそれぞれの単語について word2vec を用いて訓練した単語ベクトルのコサイン類似度を発話-応答間で計算し，top-n 組の単語の類似

度平均によって応答のスコアリングを行った．しかしnの値や類似度計算の対象とする単語の品詞の制限によらず，結果はほとんどランダム選択と変わらないものであった．

4.4.1 実験設定

実験にあたって，我々は学習・評価のために3種類のデータセットを用いた．1つはNTCIR-12 Short Text Conversation Japaneseタスク[42]において指定された500,000組のツイート・リプライIDのうち，Twitter APIを用いて取得できた421,050組からなるデータセットである（以降，NTCIR-devデータセットと呼ぶ）．2つ目は，人手評価によるテストのために同タスクから配布された，202組のツイート・リプライIDからなるデータセットである（以降，NTCIR-testデータセットと呼ぶ）．3つ目は，我々の研究室において2011年3月より継続的に収集しているTwitterのデータセットを元に構築したものである[4]．その中でも本節で行う実験で用いたものは，2011年から2013年までに収集したおよそ230,000,000組のツイート・リプライからなるデータセットである（以降，UTデータセットと呼ぶ）．

我々はまず，分散表現を用いた発話のクラスタリングによって，ツイートデータをそれぞれのサブドメインごとに分割するために，word2vec[5]を用いたskip-gramによる単語ベクトルをUTデータセットから学習した．この際，単語ベクトルの次元数は200，skip-gramにおける窓幅は5に設定している．

次に，NTCIR-devデータセットの中から選択した100,000組のツイート・リプライに対し，そのツイート部分に含まれるそれぞれの単語のベクトルを平均する事で，そのツイート全体を表すベクトルとする．このベクトルに対してscikit-learn[6]で実装されているk-meansクラスタリングを適用した．k-meansのクラスタ数については1から40までの範囲を試し，その結果として得られた各クラスタのツイートと，

[4]収集の対象とするユーザとしては，2011年3月に30名程度の著名な日本人ユーザを選択し，そのユーザのタイムラインを公式APIで継続的に収集するとともに，それらのユーザがメンション・リツイートを行ったユーザのタイムラインも収集対象として追加することで拡大していった．

[5]https://code.google.com/archive/p/word2vec/

[6]http://scikit-learn.org/stable/

それに伴うリプライを発話に対する適切な応答として考え，TensorFlow[7] を用いて実装された LSTM の訓練に用いた．また，LSTM のハイパーパラメータについては事前に NTCIR-dev データセットの中の一部を用いてチューニングを行った．次に，テストデータと評価尺度について述べる．

応答候補の順位付けに対する自動評価　テストデータとしては NTCIR-dev データセットから選んだ 1000 件のツイート・リプライの組を発話に対する正しい応答の組として選択し，その 1000 件のそれぞれに対して UT データセットからランダムに選んだ 19 応答を応答候補として加えた．そのようにして，問題となる 1000 件のツイートそれぞれに対する，正解のリプライを含んだ 20 応答を得た．

その上で，前述した学習データを用いてクラスタごとに訓練した LSTM を用いて，その応答候補の応答としての妥当さを順位付けする．評価指標としては Wu ら [43] の **1 in t P@k** を用いた．これは，t 件の応答から k 応答選択した時，正解の応答が含まれている割合を指す．本実験では，$t = 20, k = 3$ に設定した．この実験に関する結果は第 4.4.2 節で述べる．

NTCIR-12 STC タスクにおける第三者評価　テストデータについて，発話については NTCIR-test データセットの 202 件を用いた上で，それぞれの発話に対し，NTCIR-dev データセット全体から応答候補の選択を行う．この際，NTCIR-dev データセットは NTCIR-test データセットのそれぞれの発話に対して実際に行われた応答を含んでいないため，発話・応答内容を考慮しない何らかの手段で実際に行われた応答を選ぶという手段は不可能である．

また，応答候補の数が膨大であるため，第 4.2 節で述べた分類器を用いて応答候補を 421,050 件から 500 件に絞込みを行った上で，その 500 件から LSTM を用いて応答候補の順位付けを行った．このようにして得られた応答に対してそれぞれ上位 5 件を提出し，人手評価を行う．

また分類器の訓練については, NTCIR-dev データセットの中からランダムに 420,850

[7] https://www.tensorflow.org/

手法	精度 (1 in 20P@3)
ランダム選択	15.0%
ベースライン ($k = 1$)	30.8%
提案手法 ($k = 10$)	33.2%
提案手法 ($k = 20$)	35.4%
提案手法 ($k = 40$)	35.0%

表 4.1: 実験結果: 応答候補の精度 (1 in 20P@3) (k はクラスタ数)

組の発話・応答を正例として選択し，それらの発話に対する同数の応答を UT データセットから負例として選択した（合計で，841,700 組の発話・応答を用いた）．NTCIR-dev データセットのうち分類器の訓練に用いなかった 200 組については，分類器による応答の事前絞込性能の評価のために用いた．これについては NTCIR-test データセットに対する実験結果と合わせて，第 4.4.3 節で述べる．

4.4.2 応答候補の順位付けに対する自動評価

この実験では提案手法のクラスタ数 k に対し，全データを単一のモデルの上で訓練したベースライン ($k = 1$) と，k を 10, 20, 40 に設定したものとの比較を行った（表 4.1）．以降，$k = 1$ のものを single モデル，k=10, 20, 40 のものを k-cluster モデルと呼ぶ．実験結果では，いずれの場合も k-cluster モデルの精度がベースラインとなる single モデルを上回っている．

その中でも，最も良い結果が得られた 20-cluster モデルについてさらに詳細な分析を行う．表 4.2 に 20-cluster モデルの各クラスタごとの single モデルとの比較結果を，表 4.3 にベースラインと提案手法の応答選択結果の一部を示す．「サブドメイン」は，我々が各クラスタに含まれるツイートの内容から判断し，手動でラベル付けしたものである．「要素数」はそれぞれのクラスタで訓練・テストに用いられた発話-応答ペアの数であり，「正解数」では同じ 1000 問のテストケースに対して，ベースラインとなる single モデルと 20-cluster モデルの正解数を比較している．

まず，各クラスタのサイズと精度の向上率に注目すると，小さなクラスタ (〜5000) については応答のパターンが限られている事から元々の正解率が比較的高い事もあ

ID	サブドメイン (話題，単語，口調)	#要素数 train	test	#正解数 提案手法	ベースライン	精度向上 $\frac{\triangle\text{\#正解数}}{\text{\#要素数 (test)}}$
13	-	11801	108	**38**	27	10.19%
7	-	11524	124	**37**	32	4.03%
14	政治，経済，社会問題	10294	130	**48**	38	7.69%
3	-	9743	94	**32**	23	9.57%
16	アニメ・漫画	6747	56	**11**	10	1.79%
12	-	6552	66	**24**	23	1.52%
19	ゲーム	5677	50	**13**	5	16.00%
10	-	5627	45	**14**	13	2.22%
1	'!', '?'	5190	63	**17**	15	3.17%
0	眠い，辛い，愚痴	5064	52	17	**21**	-7.69%
15	-	4908	50	22	**24**	-4.00%
17	数字	3803	31	5	**7**	-6.45%
6	飲食	2630	16	**6**	4	12.50%
2	フォロー, RT ありがとう (フランク)	2252	33	29	**30**	-3.03%
18	'!!!' を含む	1869	17	**8**	**8**	0.00%
8	フォロー, RT ありがとう (丁寧)	1553	13	**12**	**12**	0.00%
4	挨拶	1537	21	**7**	6	4.76%
9	'...' を含む	1326	12	**3**	2	8.33%
5	おはようございます	1174	13	**9**	6	23.08%
11	叫び，連呼	729	6	**2**	**2**	0.00%
合計		100000	1000	354	308	4.60%

表 4.2: 提案手法 ($k = 20$) とベースライン ($k = 1$) のそれぞれのクラスタにおける結果の比較

り，精度向上はしたものの際立った変化は見られなかった．大きなクラスタ (5000〜) については軒並み精度が上昇しており，件数で比較した全体の精度の上昇への寄与も主に大きなクラスタによるものである．全体としては，20 クラスタ中 13 クラスタで精度の向上が確認された．

我々が学習データの量とモデルの精度の関係について検証するために行った予備実験によると，同じテストデータに対して，学習データ数 10,000 件の single モデルの精度は 26.1%であった．学習データ 100,000 件の single モデルと比較するとその精度は 4.7%低下している．それに対し single モデルと比較して，k-cluster モデルのそれぞれのクラスタでは 1 つのモデルあたりの学習データのサイズが 0.7%〜11.8%となっている事を考えると，おおむねサブドメインへの分割による効果はデータの減少を十分に補償していると言える．

これらの結果から，サブドメインへの分割によって自身のドメインと関連性の薄い応答が学習データから削減され，それぞれのモデルの詳密化によってより良い学

発話	応答（正解）	応答（ベースライン）	応答（提案手法）($k = 20$)
あ、見るの忘れてた。おめでとう！	ありー！ 見直してくれてありがとう！	今年は 1 年ありがとうございました	ありー！ 見直してくれてありがとう！
映画も見ました？ 僕は両方見ました！	映画も見ましたよ—	ありがとうございます！ よろしくお願いします！	映画も見ましたよ—
一言のやつ相変わらずバカなんでしょうね	それだけかい！ 笑	おかえりなさい！ お疲れ様でした	それな！ 楽しみにしてる！！
そちらの学校は楽しいの？	楽しいよ。	お疲れ様でした！	楽しいよ。
ベジータが体モノマネと対になってたってことだよね。 すごい…。	それね！ ほんとすごいですよね！	おはようございます。家に帰るのが怖いです笑	それね！ ほんとすごいですよね！
アプリのランキングにひどいやつがあって笑った	○○、16 日てご飯食べいくよね？	うわ、お大事にしてください。	たまにありますね。 あれ、なんなんでしょう。
俺も九段下の駅をおりて坂道を人の流れ追い越して行きてーなー	そこから見えるひかるたまねぎ見たことある？？	え？ ええっっ！？ おめですぅぅ！！！	そうそう！ 結構邪魔よね、あれ。
質問です！！ LINE で話したことない先輩をブロックして削除するのはやめた方がいいですか？	やめた方がいいと思うよ・・・	おつかれさまです・・・！	やめた方がいいと思うよ・・・
あれ、昨日飲みまくってなんかやらかしたかな 六万くらい消えてるまぁいいか	６万飛ばすなんてなんて大人だ・・・	うわあああああ！ なつかしい!	お疲れ様でした。良いライブになったかな？
カントリーマァムのドリンクのやつが見つかりません。	ローソン限定じゃなかったっけ？	先輩、おはよーございます♪	ローソン限定じゃなかったっけ？
ロックスターとレッドブルを連続で飲むと気持ち悪いね。 特にね。	当たり前だよね。 今日はね。 ありがとうね。	来年もよろしくお願いいたします。	アヤちゃんだね。。 癒されたよね。。

表 4.3: 提案手法によってベースラインの典型応答が改善された例 (top-1 のものを表示).

習が行われた，と説明できる．例えば，single モデルにおける応答選択の失敗例としては「おはよう」などのような高頻度な応答（典型応答）が過剰に学習された結果，不適切な状況でもそれらの応答を選択してしまうというものであった．しかし提案

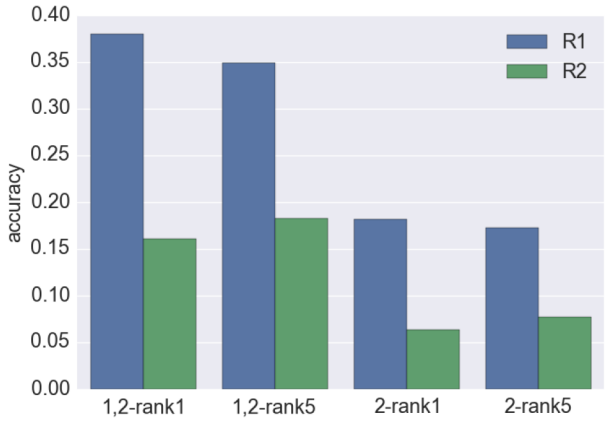

図 4.2: NTCIR-test データセットを用いた評価結果

手法によって,「おはよう」と返す事が自然な発話の大部分はサブドメインとして別のクラスタに分割される．その結果，それ以外のクラスタにおける典型応答の頻度は下がり，頻出する応答を過剰に選択してしまう問題が緩和されたと考える．一方，小さなクラスタを構成するサブドメイン，誕生日のお祝いや起床・就寝、フォロー・RT に関する挨拶，などはある種典型的な応答が存在するような発話であり，そうした発話に対する典型的な応答がより適切に学習できている，と解釈可能である．

4.4.3 NTCIR-12 STC タスクにおける人手評価

第 4.4.2 節に対し，本節で行う実験は選択された応答に対する人手評価によってシステム全体の評価を行うことを目的とするものである．その際に行われる人手評価では，選択された応答について 0 (inappropriate), 1 (appropriate in some context), and 2 (appropriate) の三段階のラベルが複数のアノテータによって付与される．

これらのラベルに対して，提出した応答の上位 n 件に対する評価結果が 2 であった割合を 2-rank$n$，1 または 2 であった割合を 1,2-rank$n$ として表す．この 2 つに対して $n = 1$ の結果と $n = 5$ の結果，計 4 通りを図 4.2 に示す．

今回我々は 2 種類の応答選択結果を提出した．R1 は第 4.1 節，第 4.2 節，第 4.3 節

図 4.3: 分類器による応答絞込性能の評価

で述べた 3 つの手法，すなわちクラスタリング，二値分類器による応答の事前絞込，LSTM による応答モデルの全てを用いた結果で，R2 は二値分類器による事前絞込のスコアが高いものをそのまま採用した結果である．

それに加えて我々は分類器によって自然絞込された応答候補がどの程度適切であるかを確認すべく，順位付けされた応答候補の内、正解の応答が上位 N 位以内に存在する割合によって分類器による応答選択性能を評価した。 NTCIR-dev データセットからランダムに選択した 200 組を正解のセットとし，それ以外の 420K(420850) 組の応答部分を応答候補としている。また、この実験では異なるサイズのデータ（8.4M, 840K）によって訓練された 2 つの分類器について比較している． 図 4.3 にその結果を示す。分類器の性能は学習データのサイズに対して良くスケールするため，より多くのデータを用いることによって質の高い応答の事前絞込が可能であると期待される一方で，大きな方の分類器であっても上位 500 位に正解のツイートが存在するのは 16%と十分とは言えない割合であり、LSTM を用いた手法の高速化を行い，現実的に計算可能な応答候補の数を向上させることが必須である。

NTCIR12-STC タスクにおける他チームとの比較結果を図 4.4 に示す．他チームの

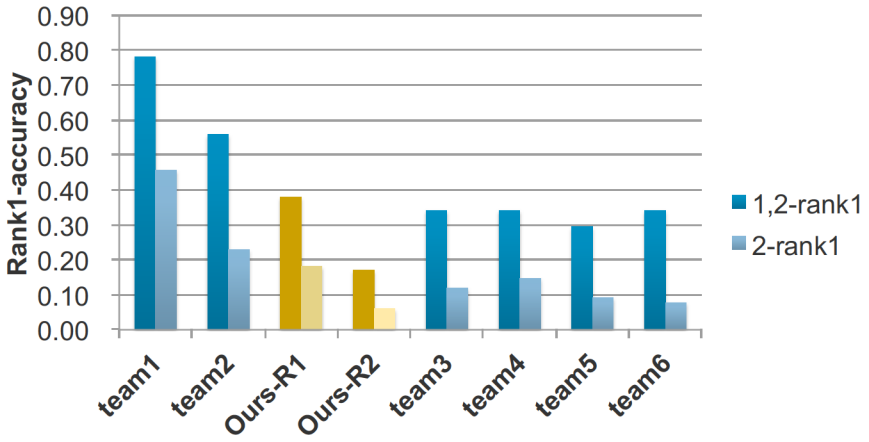

図 4.4: 結果の他チームとの比較 (1,2-rank1)

結果については複数提出したもののうち，最も数値が高いものを表示している．現実的に対話システムに許される計算時間を考慮し，全応答候補から 0.1% (500,000 応答から 500 応答) に事前絞込みを行った上での結果にも関わらず，我々の提案手法 [44] は他チームと比較しても概ね良い結果を残しており，かつ応答候補の絞込のための分類器のみを用いた結果 (R2) に対して大きく結果が改善されている．

また，他チームが用いた手法の比較についても述べる．まず team2[45], team4[46], team5 は [47] それぞれ異なる手法 (Latent Dirichlet Allocation (LDA), TF-IDF, word2vec) を用いて発話・応答文の類似度計算を行い，それらを手がかりとして応答選択を行っている．

team3[48] については同様に TF-IDf や word2vec を元に計算した類似度を用いて直接応答選択を行う手法や，類似度やその他の特徴量を手がかりとした教師あり学習によってある応答の適切さを計算する手法を用いている．

team6 は word2vec に基づく類似度を手がかりとした手法に加え，単語の共起についてのグラフ構造を元に応答選択を行う手法を用いている．

また，最も優れた結果を残した team1 は発話・応答のパターンや構造に注目したルールベースの対話モデルであったことは特筆すべきである．本タスクのように，学習データが比較的小規模であるような場合にはルールベース対話モデルは依然大

きな効果を持つ．また，こうした手法は教師あり学習に基づく場合と比べてエラー分析が容易であり，多様性を犠牲にする反面で明らかに不自然な応答は抑制される傾向にある．そのため，現在の実用的な対話システムにおいては統計的対話モデルとルールベース対話モデルとの融合 [5] を行うことによって，両者の利点を享受することが現実的であると考える．

第5章　多様な発話状況を考慮したニューラル対話モデル

第 4 章では発話内容からクラスタリングに基づく内的な発話状況の分類を行い，独立にモデルを訓練することによって応答性能が向上することを確認した．しかしその一方で，モデルごとに学習データを分割する事によって発話状況の影響が小さい，普遍的な言語的知識の学習が妨げられてしまうという問題点は未解決である．また，データサイズが小さくなることで 1 つのモデルあたりの学習時間は若干軽減されるものの，複数のモデルを学習するため計算コストが大きいという点も問題である．

そのため本章では SEQ2SEQ 対話モデル [30] をベースに，単一モデル内で大域的な情報と発話状況に応じた局所的な情報を並列に考慮する Local/GlobalSEQ2SEQ モデルを提案し，その効果の検証を行う．

5.1　提案手法

5.1.1　SEQ2SEQ 対話モデル

本節で行う実験において，我々は SEQ2SEQ 対話モデルを採用した．SEQ2SEQ 対話モデルでは発話を入力とし，まず Encoder と呼ばれる RNN によって発話の単語列を順次読み込み，発話内容を表現する実数値ベクトルに変換する．続いて，得られた実数値ベクトルを Decoder と呼ばれる RNN の初期状態とし，RNN の内部状態に基づいて単語を一単語ずつ入出力する．

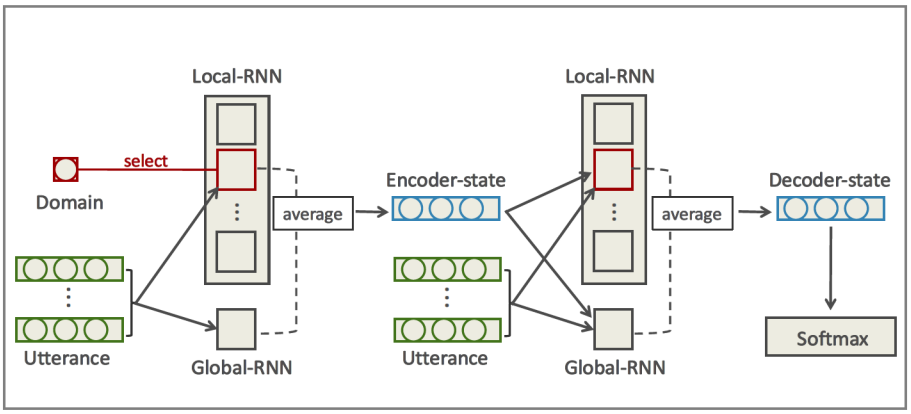

図 5.1: Local/Global SEQ2SEQ

本研究では RNN として Long-Short Term Memory (LSTM) [16] を採用し，応答選択テストの際は応答候補文に対し，各タイムステップにおけるクロスエントロピー損失の平均を擬似的なスコアとして用いて応答候補の順位付けを行う．また，ソフトマックス関数としては Jean ら [49] の sampled softmax を用いることで，高コストなソフトマックス関数における計算の高速化を行った．

5.1.2 発話状況を考慮した対話モデル

発話状況を考慮したニューラル対話モデルを訓練するにあたって，発話以外の情報をモデルで考慮する方法としては

(1) 発話状況に応じて一部のパラメタを独立に訓練する

(2) 発話状況を特徴量として外部から与える [20, 35]

の 2 通りが考えられるが，本研究では前者に基づく Local/Global SEQ2SEQ を提案する．

提案モデルでは Encoder, Decoder 共に 2 種類の RNN を同時に訓練する．1 つは発話状況によらず共通して用いる Global-RNN，もう 1 つは発話状況ごとに独立な Local-RNN であり，その 2 つの出力を平均することで全体の出力を得る (図 5.1)．ま

た Local-RNN のみを用いたモデルでの予備実験も行ったが[1]，Global-RNN を加えた場合と比べいずれも低い性能しか得られなかったため，第 5.2 節では後者の結果のみを記す．

5.1.3 着目する発話状況

対話に付随する発話状況としては様々なものがあるが，第 1.1 節で述べたように，本稿では発話内容に基づく内的な発話状況と，発話時間・場所や発話者間の人間関係などに基づく外的な発話状況に分けて，それぞれ用いることを検討する．提案モデルは発話状況が所与であると仮定するため，それぞれについて教師無しで（安価に）得られる発話状況として，以下を用いる．

content　内的な発話状況については第 4 章に倣って発話内容を実数値ベクトルで表現し，それをクラスタリングすることで得る．具体的には，対話モデルの学習データを用いて word2vec[2] で単語のベクトル表現を得た後，発話中の単語のベクトル表現を平均することで発話のベクトル表現を得る．得られた発話ベクトル表現を yakmo[3] による k-means クラスタリングを用いて分割・分類し，得られたクラスタを発話状況とみなす．実験では，発話状況の分割数 k は 10 に固定した．

season/month/hour-n　外的な発話状況については，今回のデータセットにおいて自明に利用できるタイムスタンプを用いる．具体的には発話のタイムスタンプを元に 1-12 月の 12 種類（月ごと）への分割と，3-5 月, 6-8 月, 9-11 月, 12-2 月の 4 種類（春夏秋冬）への分割を行った．分割数ごとに季節単位で分けたものを season，月単位で分けたものを month と呼ぶ．

また朝や夜といった発話時の時間も応答に影響すると考えられる事から，同じく外的な発話状況として採用した．具体的には season/month と同様のタイムスタンプ

[1] embedding/softmax layer を共有しているという違いはあるが，第 4.3 節とほぼ同一のモデルである．

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

[3] http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/yakmo/

を用いて，その時刻について分割を行った．分割は 0 時から 6 時間ごとに 4 種類への分割と，同じく 0 時から 3 時間ごとに 8 種類への分割を行った．それらについては hour-n と呼ぶ．

本稿では人間関係や場所のような，現在用いるマイクロブログの対話データでは陽に与えられず，推定が必要な外的な発話状況の利用については今後の課題とする．

5.2 実験

提案手法の有効性を検証するため，応答選択タスクによって手法の評価を行う．

5.2.1 設定

学習・評価データ 学習・評価のためのデータセットとしては，我々の研究室において 2011 年 3 月より継続的に収集している Twitter のデータ (第 4.4.1 節で述べた UT-データセットの構築元データを更新したもの) から構築した．具体的には，各投稿（ツイート）を発話とみなし，あるツイートとそれに対するリプライを 1 組の発話・応答ペアとした上で，学習データとして 2014 年から約 23,563,865 組，テストデータとして 2015 年の各月から 500 組ずつ合計 6,000 組を抽出し，それに加えてダミー応答候補として同年のツイートからランダムに 114,000 応答を抽出した．続いて MeCab を用いて各発話・応答を形態素に分割した．テストに用いたデータはすべて 20 単語以下のものに制限している．また発話は他のツイートに対するリプライとなっていないものに限定することで，リプライの連鎖の中に存在する文脈の不足によって回答不能となるケースを可能な限り減らしている．

488

比較モデル 実験では以下の 4 つのモデルを比較する．

Baseline(SEQ2SEQ) [19] 第 2.5 節参照.

Local/Global SEQ2SEQ (提案手法) 第 5.1.2 節参照.

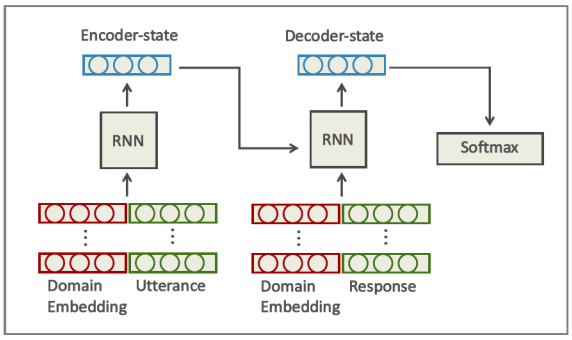

図 5.2: Domain-Embedding

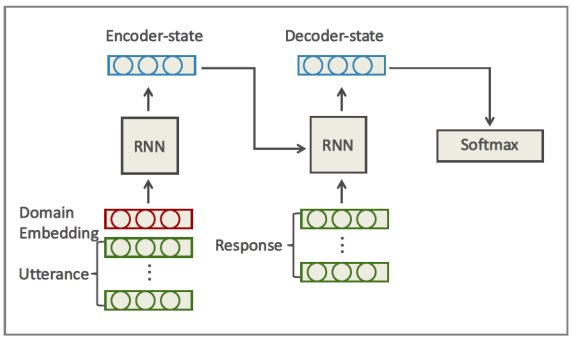

図 5.3: Domain-Header

Domain-Embedding [20] 入力に加えて発話状況を表現するベクトル (Domain-Embedding) を RNN の各時刻の入力と結合させる (図 5.2). この Embedding についても，応答の最適化から同時学習が行われる.

Domain-Header [35] 発話を構成する単語列の先頭にその発話の発話状況を表す ID を加える (図 5.3). 発話状況の ID は他の単語と同様に Embedding として入力され，応答の最適化の際に同時学習される.

以上のモデルを TensorFlow[4] を用いて実装した. RNN は 3 層の多層 LSTM を採用し，最適化は Adam [50] によって行った. 主要なハイパーパラメータは語彙: 100,000 word，隠れ層: 200 次元，dropout 率: 0.25，初期学習率:1e-5 に設定した.

[4] https://www.tensorflow.org/

評価手順　前述した学習データを用いて訓練した各モデルについて応答選択テストを行う．具体的にはあるツイートに対し，実際に行われたリプライとダミー応答候補から最大 19 応答，合計最大 20 応答候補から応答としての妥当さを順位付けする．評価尺度としては Wu ら [43] の **1 in t P@k** を用いる．これは，t 個の応答候補の中から k 応答選択し，その中に実際に行われたリプライが含まれている割合を意味する．t と k の値については，同じく Wu らの研究にならって **2P@1, 5P@1, 5P@2**, また第 4 章における実験と同様に **20P@3** を用いた．**5P@2, 20P@3** において k の値を 1 より大きなものとする，つまり正解として複数候補選択することを許す理由としては応答候補の数が増えた際に，正解の（実際に行われた）応答では無いが人間の感覚では同様に適切な応答であると考えられるものが存在する可能性を考慮しているためである．

5.2.2　結果と考察

第 5.1.3 節で述べた発話状況に対し，第 5.2.1 節で述べた 4 つの手法を用いて実験を行った (表 5.1〜表 5.5)．結果，我々が提案する Local/Global SEQ2SEQ がベースラインの結果を上回り，平均的に最も良い応答性能を示した．その理由としては Domain-Embedding, Domain-Header は発話状況を表現するベクトルを入力するモデルであるため，Embedding が入力系列もしくは RNN 内部で入力とともに繰り返し行列演算が行われるにつれ，その影響が低くなってしまうからであると考えられる．こうした発話状況の情報を Embedding として与えることの大きな利点の 1 つとして，それぞれの発話状況がベクトルとして表現されることで，類似する発話状況が類似するベクトルとして表現される，という点が挙げられる．本実験においては発話状況の分割数は最小で 4 分割，最大で 12 分割に設定したが，これらの比較手法は例えば Li らが行ったようなユーザタイプによる分割 [20] のような，分割数が多くなると予想される場合において，発話状況間の類似を効果的に学習可能であると考えられる．

また month，hour-4 をはじめとして，比較手法が提案手法を上回っている結果も

散見される．このような結果が得られた原因として，主に次のような理由が考えられる．提案手法である local/global SEQ2SEQ は発話状況ごとに独立に訓練したモデルを併用することで，発話状況を考慮するモデルであることを第 5.1.2 節において述べた．このモデルにおいては，学習データが減少してしまう問題を大域的なモデルを併用することによって軽減しているものの，局所的なモデルについては依然発話状況ごとに独立なデータによって訓練される．そのため細かく発話状況を分割したとしても，それぞれの発話状況下における発話・応答の変化が十分に得られない場合，学習データの減少によるデメリットの方が強くなる．こうした理由から，例えば local/global SEQ2SEQ の season(4 分割) と month(12 分割) の結果を比較すると，月の違いによる発話・応答の変化は季節の違いほど大きなものではなかったため，分割することがマイナスに働いた例であると考える．

以上を総括すると，発話状況の差異による発話・応答への影響が大きく分割数が比較的少ない場合は local/global SEQ2SEQ，その逆の場合は比較手法のいずれかを用いることが適当であると考える．

次に，表 5.6 に各手法の応答例を示す．発話状況については season，評価尺度は 1 in 5P@1 の結果から引用した．Li らが述べるように [51]，マイクロブログなどの雑談対話コーパスを用いて訓練した一般的なニューラル対話システムでは典型的な応答が非常によく見られることが問題となる．事実，我々がベースラインとして用いた通常の SEQ2SEQ モデルにおいても「おつかれ」や「ごくろうさま」をはじめとする典型的な応答が散見された．

この現象は訓練データ中で典型応答の頻度自体が高いこと，また機械翻訳のような入出力間で概ね 1 対 1 の対応が取れている場合と異なり，雑談対話においては発話状況のような制約無しには発話に対する応答の自由度が高すぎるため，汎用的な応答を選びがちであるということに起因すると考えられる．一方で提案手法や比較手法においては，発話状況がある種の制約として働くことによって，典型応答の出現が抑制された．

あるテストケースに季節性が存在するか否かについては主観的なものになるため季節性がどこまで結果に反映されているかについての定量的な評価は難しいが，テ

表 5.1: season: 1 in t P@k

Model	1 in 2P@1	1 in 5P@1	1 in 5P@2	1 in 20P@3
Baseline	66.7%	37.8%	60.1%	32.7%
Local/Global SEQ2SEQ	**70.2%**	**39.9%**	**64.2%**	**35.5%**
Domain-Embedding	*65.4%*	36.3%	*58.0%*	31.7%
Domain-Header	*69.1%*	*39.3%*	*62.5%*	*35.2%*

表 5.2: month: 1 in t P@k

Model	1 in 2P@1	1 in 5P@1	1 in 5P@2	1 in 20P@3
Baseline	66.7%	37.8%	60.1%	32.7%
Local/Global SEQ2SEQ	68.4%	38.9%	62.9%	34.1%
Domain-Embedding	**69.6%**	**40.0%**	**63.4%**	**34.6%**
Domain-Header	65.7%	36.6%	58.8%	32.3%

表 5.3: content: 1 in t P@k

Model	1 in 2P@1	1 in 5P@1	1 in 5P@2	1 in 20P@3
Baseline	66.7%	37.8%	60.1%	32.7%
Local/Global SEQ2SEQ	**69.0%**	**39.5%**	**62.5%**	**35.4%**
Domain-Embedding	67.3%	37.7%	60.8%	33.1%
Domain-Header	68.2%	38.6%	61.4%	33.2%

表 5.4: hour-4: 1 in t P@k

Model	1 in 2P@1	1 in 5P@1	1 in 5P@2	1 in 20P@3
Baseline	66.7%	37.8%	60.1%	32.7%
Local/Global SEQ2SEQ	67.8%	38.2%	61.5%	33.4%
Domain-Embedding	**70.0%**	**40.3%**	**63.6%**	**35.8%**
Domain-Header	68.9%	39.9%	62.3%	33.4%

スト結果を俯瞰した結果全体としては季節性があると考えられる例についての応答性能は向上していた．一方で，発話状況が「夏」の例では 正解の応答は「寒そう」という夏にはそぐわないと考えられる表現である．このように，発話状況を考慮することで逆に応答に失敗するような例も存在したことは興味深い．

表 5.5: hour-8: 1 in t P@k

Model	1 in 2P@1	1 in 5P@1	1 in 5P@2	1 in 20P@3
Baseline	66.7%	37.8%	60.1%	32.7%
Local/Global SEQ2SEQ	68.5%	**39.8%**	61.9%	**34.6%**
Domain-Embedding	68.6%	39.1%	62.3%	34.4%
Domain-Header	**69.1%**	39.5%	**62.6%**	34.1%

表 5.6: season, 1 in 5P@1 における応答例 (太字が正解応答)

Domain	春
Utterance	寒いと思ったら雪降ってた
Baseline	おつかれ！
local/global SEQ2SEQ	**まじ?この時期にやめてほしいサクラサケ**
Domain-Embedding	多分なにも考えてない。こいつはそうだ
Domain-Header	おつかれ！
Domain	夏
Utterance	アイス界に行ってくる
Baseline	**寒そう**
local/global SEQ2SEQ)	マチキンもあっただろ!
Domain-Embedding	なんでこんなにも起きるのが遅いんですか?
Domain-Header	呼ばれてないけど春ですよ!!
Domain	秋
Utterance	秋の花粉来てるなあ
Baseline	ご苦労だった、ゆうこ。
local/global SEQ2SEQ)	**もうかなり前から鼻ヤバイです…**
Domain-Embedding	ご苦労だった、ゆうこ。
Domain-Header	関西以西だと思いますけど、最近は標準語化してるかもですねー
Domain	冬
Utterance	冬らしからぬ紅い髪で除雪するエレノア
Baseline	あいこん
local/global SEQ2SEQ)	**冬は白髪になろう**
Domain-Embedding	**冬は白髪になろう**
Domain-Header	**冬は白髪になろう**

第6章　おわりに

6.1　本研究のまとめ

本研究では対話応答タスクにおいて，発話内容に加えてその背後に存在する発話状況を考慮した応答を可能にすることを目的とし，(1) 明示的に存在せず，発話内容から推測を行う事で得られる内的な発話状況を考慮する対話モデルの提案と (2) 発話状況を考慮することによる学習データの減少を解決する対話モデルの提案を行った．

まず，我々は事前に学習を行った word2vec を用いて発話のベクトル化を行った後，k-means を用いて発話内容のクラスタリングを行うことによって，内的な発話状況を得た．そうして得られた発話状況ごとに独立にモデルの学習を行うことで，応答性能が向上することを応答選択タスクによって確認した．

また，Seq2Seq 対話モデルをベースに，大域的な情報と発話状況に応じた局所的な情報を並列に考慮する Local/Global SEQ2SEQ モデルの提案を行った．その上で (1) で述べた内的な発話状況とデータから自明に取得可能な外的な発話状況を考慮した際の応答性能への影響について応答選択タスクによって既存のニューラル対話モデルとの比較を行った．その結果，ベースラインや比較手法と比べて提案手法が平均的に最も良い結果を得られた．

第7章　今後の課題

本研究で行った実験において，提案手法による対話モデルの応答性能の向上が確認出来たものの，表 5.1〜表 5.5 に示すようにその応答性能は決して高いとは言えず多くの改善の余地が残る．本研究において行った実験結果，特に応答に失敗した例からの分析結果をもとに，今後の課題点について以下に述べる．

7.1　学習データから陽に取得不可能な発話状況の利用

本研究で対象としたもの以外にも，人間関係や発話者，応答者の性質など，より発話・応答に影響しうると考えられる発話状況は数多く存在する．しかし，そうした情報は発話・応答を通して間接的にのみ表れる．そのため，そうした発話状況を直接データとして取得することがほとんどの場合不可能であり，かつ人手によるアノテーションのコストも高いことは非常に重要な問題であると考える．

これらの解決法としては，1 つは事前に発話状況ラベルを設定し，人手によるアノテーションを行った小規模なデータをもとに教師あり学習を行うという方法が挙げられる．または本研究や Li らの研究 [20] で用いた，クラスタリングをはじめとした教師なしによる発話状況の分類手法の改善，もしくは複数の対話システム間で会話のシミュレーションを行わせることによって，擬似的な人間関係や発話・応答者の性質を構築するようなアプローチも有望であると考える．

7.2 語の意味理解

また，もう 1 つの大きな問題点としては，現状の語の意味表現は大規模なデータを用いて分布仮説から学習を行ったものであるため，学習データにおける低頻度語や未知語についてはその意味を捉えた分散表現を獲得することが困難である．その結果，発話において低頻度語・未知語が登場した場合，その意味を読み取ることが出来ず「こんにちは」や「それな」といった，どんな発話にも無難に対応可能である応答が頻発し，また低頻度語を用いた応答を行なうことが困難である原因の 1 つとなっていると考える．

機械翻訳タスクにおいても低頻度語，未知語の問題についての研究は行われている [52] が，分散表現を手がかりとして意味の近い単語に置換するという手法が主流であるため，粒度の細かい意味の差異が重要となったとき，大域的な意味は共通しているが局所的な意味では異なる単語を同一としてみなしてしまうという，分散表現に基づく手法における問題は依然解決されないと考える．

この問題に対する解決法として，ニューラル対話モデルにおいて一般的な手法である，事前に定めた n 語を固定の語彙とし，その分散表現を分布仮説に基づき学習するというアプローチに加え，局所的な語彙的知識をその文脈や外部知識から動的に構築することで，未知語や意味の曖昧な語であってもその意味を推測する事が可能になると考えている．

7.3 視覚的情報の利用

我々の知る限り，対話応答タスクに関するほとんどの既存研究の手法は自然言語を手がかりとして応答を行う．しかし例えば雑談対話において今見ているものについての会話をする場合であったり，タスク指向型対話システムにおいて画像をもとに情報検索を行なう場合など，言語情報に付随する視覚的画像を活用することによる応答性能向上の可能性は決して少なくないと考える．

以上で述べた課題点について，著者が博士課程進学後に取り組む予定である．

参考文献

[1] Stephanie Young, Milica Gasic, Blaise Thomson, and John D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of IEEE*, 101(5):1160–1179, 2013.

[2] Timothy W Bickmore and Rosalind W Picard. Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(2):293–327, 2005.

[3] Alan Ritter, Colin Cherry, and William B Dolan. Data-driven response generation in social media. In *Proceedings of EMNLP*, pages 583–593, 2011.

[4] Ryuichiro Higashinaka, Nozomi Kobayashi, Toru Hirano, Chiaki Miyazaki, Toyomi Meguro, Toshiro Makino, and Yoshihiro Matsuo. Syntactic filtering and content-based retrieval of twitter sentences for the generation of system utterances in dialogue systems. *Proceedings of IWSDS*, pages 113–123, 2014.

[5] 目黒豊美, 杉山弘晃, 東中竜一郎, and 南泰浩. ルールベース発話生成と統計的発話生成の融合に基づく対話システムの構築. **人工知能学会全国大会論文集**, 28:1–4, 2014.

[6] Takayuki Hasegawa, Nobuhiro Kaji, Naoki Yoshinaga, and Masashi Toyoda. Predicting and eliciting addressee's emotion in online dialogue. In *ACL*, pages 964–972, 2013.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[9] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, September 2015.

[10] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *Proceedings of ACL-IJCNLP*, pages 1577–1586, July 2015.

[11] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

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

[13] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proceedings of INTERSPEECH*, volume 2, page 3, 2010.

[14] Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*, 2015.

[15] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. A character-aware encoder for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3063–3070, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

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

[17] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[18] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[19] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[20] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany, August 2016. Association for Computational Linguistics.

[21] Geoffrey E Hinton. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

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

[23] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Hlt-naacl*, volume 13, pages 746–751, 2013.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in NIPS*, pages 3111–3119, 2013.

[26] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[27] Sascha Rothe and Hinrich Schütze. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*, 2015.

[28] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[29] Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274, Berlin, Germany, August 2016. Association for Computational Linguistics.

[30] Oriol Vinyals and Quoc Le. A neural conversational model. In *Proceedings of ICML*, 2015.

[31] Ryuichiro Higashinaka, Noriaki Kawamae, Kugatsu Sadamitsu, Yasuhiro Minami, Toyomi Meguro, Kohji Dohsaka, and Hirohito Inagaki. Unsupervised clustering of utterances using non-parametric bayesian methods. In *INTERSPEECH*, pages 2081–2084. Citeseer, 2011.

[32] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[33] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard M Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *ACL (1)*, pages 1370–1380. Citeseer, 2014.

[34] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[35] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*, 2016.

[36] Hirofumi Yamamoto and Eiichiro Sumita. Bilingual cluster based models for statistical machine translation. In *Proceedings of EMNLP-CoNLL*, pages 514–523, June 2007.

[37] Hal Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, June 2007.

[38] Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. Frustratingly easy neural domain adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 387–396, December 2016.

[39] Annie Louis, Aravind Joshi, and Ani Nenkova. Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference*, pages 147–156, Tokyo, Japan, September 2010. Association for Computational Linguistics.

[40] Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 823–833, Berlin, Germany, August 2016. Association for Computational Linguistics.

[41] Naoki Yoshinaga and Masaru Kitsuregawa. Kernel slicing: Scalable online training with conjunctive features. In *Proceedings of COLING*, pages 1245–1253, 2010.

[42] Lifeng Shang, Tetsuya Sakai, Zhengdong Lu, Hang Li, Ryuichiro Higashinaka, and Yusuke Miyao. Overview of the ntcir-12 short text conversation task. In *Proceedings of NTCIR-12*, 2016.

[43] Bowen Wu, Baoxun Wang, and Hui Xue. Ranking responses oriented to conversational relevance in chat-bots. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 652–662, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[44] Shoetsu Sato, Shonosuke Ishiwatari, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. UT dialogue system at NTCIR-12 STC. In *Proceedings of NTCIR-12*, pages 518–522, 2016.

[45] Sota Matsumoto and Masahiro Araki. Scoring of response based on suitability of dialogue-act and content similarity. In *Proceedings of NTCIR-12*, pages 515–517, 2016.

[46] Hiroshi Ueno, Takuya Yabuki, and Masashi Inoue. Yuila at the ntcir-12 short text challenge: Combining twitter data with dialogue system logs. In *Proceedings of NTCIR-12*, pages 554–557, 2016.

[47] Hiroaki Sugiyama. Selection based on sentence similarities and dialogue breakdown detection on ntcir-12 stc task. In *Proceedings of NTCIR-12*, pages 552–553, 2016.

[48] Kozo Chikai and Yuki Arase. Analysis of similarity measures between short text for the ntcir-12 short text conversation task. In *Proceedings of NTCIR-12*, pages 523–530, 2016.

[49] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1–10, 2015.

[50] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL-HLT*, pages 110–119, 2016.

[52] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 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, August 2016. Association for Computational Linguistics.

発表文献

査読なし国際会議: 口頭・ポスター発表

1. Shoetsu Sato, Shonosuke Ishiwatari, Naoki Yoshinaga, Masashi Toyoda, Masaru Kitsuregawa, UT Dialogue System at NTCIR-12 STC, The 12th NTCIR Conference on short text conversation (NTCIR-12 STC), Tokyo, June 2016.

査読なし国内会議: 口頭発表

1. 佐藤翔悦, 石渡祥之佑, 吉永直樹, 豊田正史, 喜連川優, 発話状況を意識したオンライン上の対話における応答選択, 第 30 回人工知能学会全国大会 (JSAI2016), 福岡,2016 年 6 月.

2. 佐藤翔悦, 吉永直樹, 豊田正史, 喜連川優 , 暗黙の発話状況を考慮したニューラル対話モデル, 言語処理学会第 23 回年次大会 (NLP2017), 東京, 2017 年 3 月. (発表予定)