大学院博士前期課程修士学位論文

題　　目

会話ドメインと感情を考慮したニューラル対話モデルの構築

指導教員
鬼塚真

報告者
近井厚三

2018 年 2 月 15 日

大阪大学 大学院情報科学研究科
マルチメディア工学専攻

大学院博士前期課程修士学位論文

会話ドメインと感情を考慮したニューラル対話モデルの構築

近井厚三

内容梗概

近年，音声案内や対話ロボットなど人間との会話を行うシステムが普及している．その様なシステムの一つとして，Microsoft のりんなや Apple の Siri のような雑談を行う雑談対話システムが挙げられる．雑談対話システムの実装はルールベースによる検索型の実装が主流であったが，昨今の深層学習の発展により，応答文を生成するニューラル対話モデルが注目を集めている．応答を自動生成することで，ルールベースでは返答できないような入力に対しても柔軟な応答を可能にする．しかし，深層学習による発話生成では，汎用的で固定的な発話しか応答を生成しない問題が存在する．汎用的な対話とは，「そうですね」や「確かに」等の多くの状況下の返答として当てはまりうる発話を指し，特定のドメインに即した発話生成が困難である．固定的な対話とは，一つの発話には一つの応答しか返せないことを指し，感情に応じた発話を生成することができない．既存研究では，会話のドメイン情報や，感情語彙による制御について個別に議論されているが，会話のドメインと感情制御の二つを考慮している研究は存在しない．そこで，本報告では入力発話と応答に持たせたい感情ラベルが与えられた時に，会話ドメインを自動で推定し，指定された感情に沿った応答を生成・出力するシステムを実装した．具体的には，発話生成システムである Sequence-to-Sequence Model に，会話ドメインを推定する tweet2vec ，感情を制御する Speaker Model 及び External Memory を付加した end-to-end のモデルを構築した．実装したシステムを使用し，実際にシステムが与えられた発話に対し，会話ドメインと感情を考慮した発話が生成されているかを検証した．結果として，本提案手法は既存手法に比べて流暢性と一貫性を保ったまま，会話ドメイン整合性と感情の豊かさを出力に表現することができることが分かった．既存手法では汎用的な応答生成がなされているが，提案手法ではドメイン依存の単語や感情ラベルに従った感情語彙を出力できる．一方で，ドメイン依存の低頻度語が入力に含まれる場合や感情語彙の選定に誤りがあった場合，誤った文法や感情表現が出力として生成されてしまう問題点が明らかとなった．今後はドメイン依存の低頻度語の学習を強化することや，より正確な感情語彙の選定方法を考慮する必要がある．

キーワード

ニューラル対話モデル，感情制御，ドメイン推定，Sequence-to-Sequence

目 次

1　序論

近年，特に需要が高まっている媒体の一つとして SNS (Social Networking Service) がある．SNS は他者とのコミュニケーションをとる事が主な目的の媒体であり，ソーシャルメディアの一種である．図 1.1 に日本人 SNS 利用者数の年代順にまとめたグラフを示す．グラフは ICT 総研[1] が 2017 年度の 10 月 11 日に SNS 利用動向に関する調査結果をまとめたものの一つである．

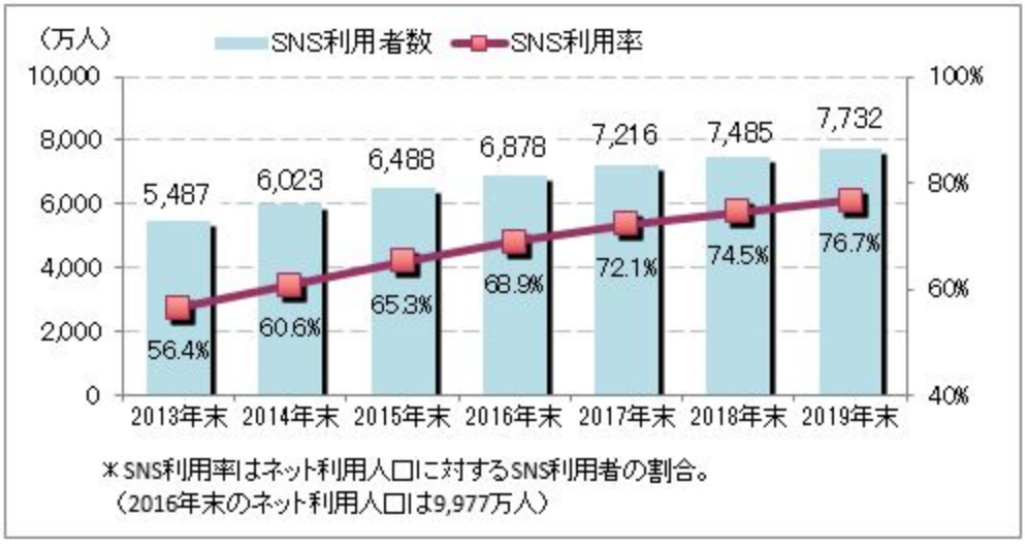

図 1.1: 日本における SNS の利用者数（ICT 総研[2] より引用）

図 1.1 からわかる通り，日本での SNS 利用者数は年々増加の一途を辿っており，2017 年末には 7,216 万人に達する見込みである．2016 年末の国内ネットユーザーは 9,977 万人と推定されるが，SNS 利用者はそのうちの 68.9 %にあたる 6,878 万人だった．2017 年の年間純増者数は 338 万人となる見込みで，１ヶ月平均で約 28 万人の利用者が増加を続けている．このことから，SNS は日本社会において欠かせない情報媒体となりつつあることが分かる．また，SNS 利用者は元々 10 代〜 20 代の若年層が多かったが，SNS 利用が当たり前になってきたことで 40〜50 代以上の年齢層にも拡大しており，登録者数・利用者数共に増加傾向が見られる．このまま普及が進んだ場合，2019 年末には利用者数は 7,732 万人，ネットユーザー全体に占める利用率は 76.7 %に達する見通しである．

上記のように年々ユーザ数を伸ばしている SNS であるが，総務省情報通信政策研究所[3] の調査では，日本国内では主に LINE[4]，Twitter[5]，Facebook[6] の三つの SNS サービスの利

[1] http://ictr.co.jp/
[2] http://ictr.co.jp/report/20171011.html
[3] http://www.soumu.go.jp/iicp/
[4] http://line.me/ja/
[5] https://twitter.com/
[6] https://ja-jp.facebook.com/

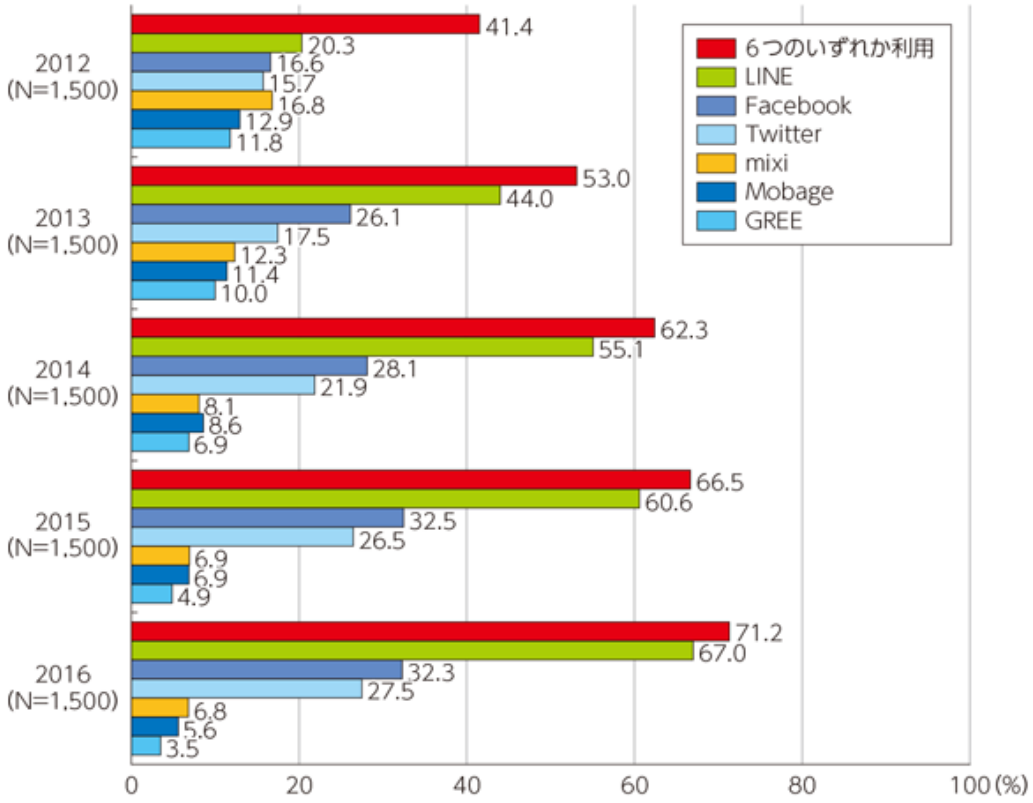

図 1.2: 代表的 SNS の利用率の推移（総務省情報通信政策研究所「情報通信メディアの利用時間と情報行動に関する調査」[8] より引用）

用率が上昇しているという結果が提出されている．SNS 利用率上昇の参考として，国内の主なソーシャルメディアの利用率を表したグラフを図 1.2 に示す． 図 1.2 によると，他の国内発 SNS を抑え，海外発サービスである LINE，Twitter，Facebook の三つの利用率が年を追う毎に増加していることがわかる．

本報告では，会話データとしての利用価値があり，会話にトピックや感情が含まれやすい Twitter や Facebook 等のマイクロブログに注目する．マイクロブログとは，ブログサービスの一種であり，利用者は現在の心境や状況，雑記等の短文をウェブサイトに投稿することができる．またユーザ間のコミュニケーションも自由に取ることができる．このようなシステムにおいて，ユーザが発する情報は基本的に現在起こっている出来事や思っている感情であることが多い．従って，この膨大なデータを収集・分析することで，現在の情勢や流行等をテキストで捉えることができる．具体例として，NHK の番組 NEWS WEB[9] のコーナーのひとつである “つぶやきビックデータ ” が挙げられる．このコーナーでは，NTT データ [10] が解析したデータから，前日に比べてツイートされた数が急増した単語を紹介する．このように，集めたテキストデータを分析する事で，現在のトレンド・

[8] http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/html/nc111130.html
[9] http://www3.nhk.or.jp/news/newsweb/
[10] http://www.nttdata.com/jp/ja/index.html

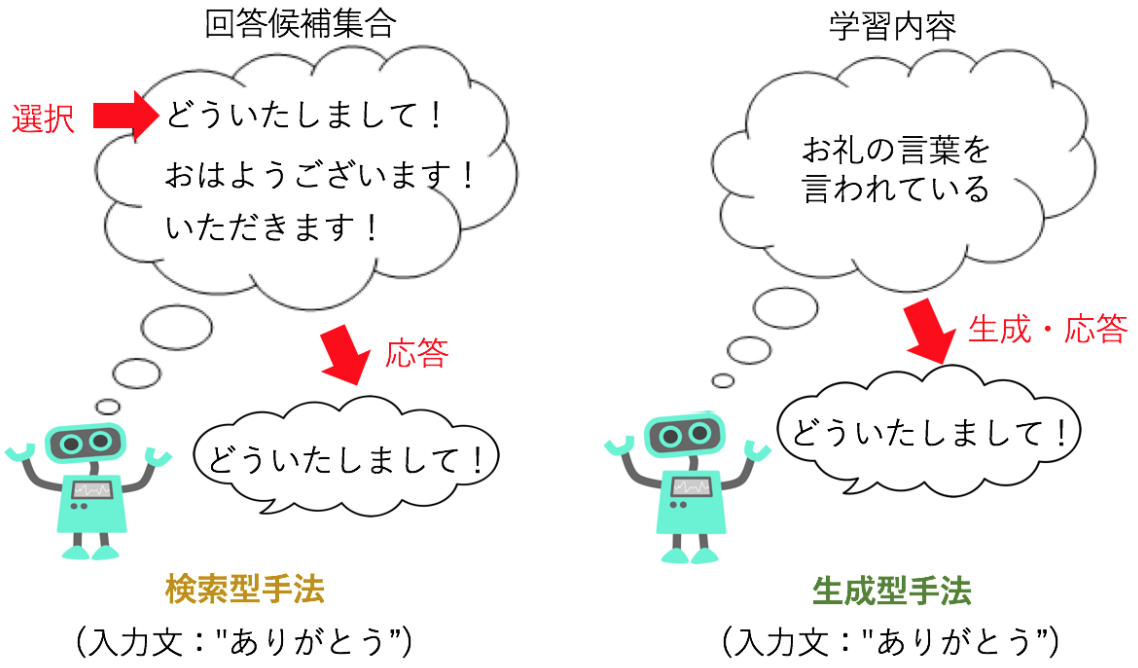

図 1.3: 対話システムにおける検索型手法と生成型手法の違い：検索型手法では発話と類似している文をデータから検索して，予め保有している返答を決定する．一方生成型手法では発話の内容を学習を元に推測し，それに基づいた応答を単語もしくは文字単位で生成する．

トピックを調べる事が可能となる．

マイクロブログのデータの利用先として，トレンド分析だけでなく，そのトレンドを加味した対話システムへの応用が考えられる．なぜならマイクロブログにおいてユーザが発する情報は出来事だけでなく，他のユーザの投稿 (post) に対して反応・返信したテキスト (reply) も含まれているからである．本報告では，そのような post と reply の文対を対話コーパスと定義する．対話コーパスのような，入力発話とそれに対する応答を記録することで，自動対話システムを構築することができる．近年の対話システムの実装用データとして Twitter データが用いられることが多い．対話システムには実装手法として検索型の手法と生成型の手法の二種類が存在する（図 1.3）．検索型の手法は，コーパスから適切な手法を検索するシステムであり，主に類似文検索や IR，知識ベースの技術が使われることが多い．検索型手法の利点として，基本的にはルールベースなので実装が容易であり，またコーパス内にある文を応答として出力するので，文法が崩れないことが挙げられる．短所としては，最適な応答がコーパス内に存在しない，もしくは応答候補が多すぎて適切な応答の判定が困難な場合，適切な応答が出力できない点が挙げられる．検索型手法に対し，生成型手法は入力発話に対して適切な応答文を機械学習等を用いて生成するモデルである．生成型手法は，検索型手法に比べて必ずしもコーパス内の文を応答する訳

では無いので柔軟な応答が可能であり，検索時間が掛からないので出力が高速であることが多い．しかし，柔軟な応答を生成するには，一般的にはコーパスのデータ量に依存しており，また，応答文の文構造が検索型手法に比べて崩れやすい．

昨今，深層学習技術の発展により，対話システムにもニューラルネットを用いた手法が用いられており，ニューラルネットで応答文を生成するニューラル対話システムが生成型手法の最たる例である．2018 年現在，広く知られているニューラル対話システムとして Sequence-to-Sequence (seq2seq) [1] が挙げられる．seq2seq は，入力発話情報を扱うエンコーダと，出力を制御するデコーダの二つによって構成されたモデルであり，対話システムのみならず，機械翻訳やその他シーケンスを入力としたシステムへの応用が為されている．本報告では，この seq2seq によるニューラル対話システムに注目する．seq2seq は生成型対話システムとして，コーパスにない発話を入力としても，出力シーケンスを予測して柔軟な応答がなされると考えられる．しかし，seq2seq の出力単語はそれ以前の出力単語群に従って出力されるので，入力が複雑な文である場合，同一かつ簡易的な文言でしか返答しないことが報告されている [2, 3] (Blandness Problem)．特に，対話分野においては，対話システムは会話のトピックを意識し，かつ，主観的な感情を含んだ発話を生成することがより人間らしいシステムになると考えられる．現在，対話分野で上記の問題に取り組んだ論文は複数存在するが，一度に会話ドメインと感情を考慮した研究は存在しない．

そこで本報告では，会話ドメインと感情制御を同時に考慮した対話システムを提案する．実際の Twitter 及び Facebook のテキストデータを用いて，上記に述べたような対話システムを構築し，その出力を既存手法である seq2seq の出力と比較し，その流暢性や応答の一貫性，会話ドメイン整合性，感情の豊かさの観点から人手評価することで，本提案システムが応答に与える影響を分析する．

2　関連研究

本章では，本報告に関連している既存研究について述べる．本報告では単語や文書をベクトルで表現することで，テキスト情報を取り扱っている．従って，現在の言語処理分野での単語や文書のベクトルへの変換手法について述べる．次に，本報告の感情制御と深い関わりのある感情分析について説明する．最後に対話システムの研究，特に Sequence-to-Sequence Model を用いたニューラル対話モデルをメインに関連研究を紹介する．

2.1　単語・文書のベクトル化に関する研究

通常，文書をコンピュータで取り扱う際，文書を何らかのベクトル表現に変換することが一般的な手法ある．つまり，大量の文書群から単一の辞書を作成し (単語の ID 化) ，そこから文書をベクトル化する．その際，情報検索に使われる技術の一つとして，TF-IDF と呼ばれるベクトル内の各単語の重み付け技術が存在する．これは，単語の出現頻度を考慮したモデルとなっており，例えば “私” という単語は一般的であり，多くの文書で登場するので，その文書の特徴を表す要素になりづらい (従って，重みを小さくする) という考えに基づいている．このような技術を用いることで，より類似した文書の特徴を捉えることができる．

また，単語の周辺共起から文書群のトピックを推定する技術 [4][5][6] も存在しており，これらを総称して，トピックモデルと呼ばれる (潜在意味解析とも呼ばれる) ．トピックモデルの根幹にあるものは，数学の行列分解 [7][8][9] を使用しており，これに言語学的な情報の意味付け (統計学における分布仮説) を行ったものであると解釈される．

近年では，単語の出現をニューラルネットによって学習させるような技術が使われており，これによって得られる特徴ベクトルを分散表現と呼ばれる．Mikolov Tomas ら [10][11][12][13] が提案した word2vec や，Jeffrey Pennington ら [14] が提案した GloVe が有名である．上記二つの技法は主に単語をベクトル化する技法であるが，これに文書単位を拡張させたようなモデルの一つとして doc2vec (Paragraph Vector)[15] が挙げられる．これらの手法は，後に提案される seq2seq [1] に大きく影響を与えた手法である．

2.2　感情分析に関する研究

2.2.1　感情語彙を用いた感情分析

言語処理における感情分析の研究は，チャットや Web 掲示板の普及につれて，人間の感情を含んだテキストデータが多く出現したことから，その必要性が高まっている．感情分析において，最も使用されるデータは Sentiment Lexicon である．これは，“happy” や “sad” といった直接感情を表す単語を語彙情報として考え，その情報を基に感情分析することが一般的である．Turney (2002) [16] は，映画のレビューを positive と negative の

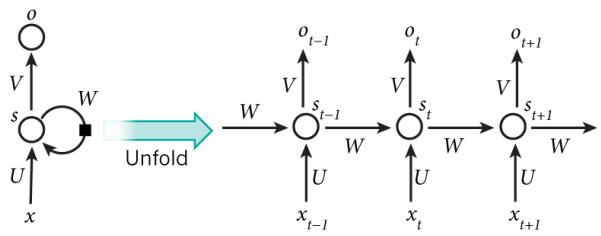

図 2.1: Recurrent Neural Network（[19] より引用）

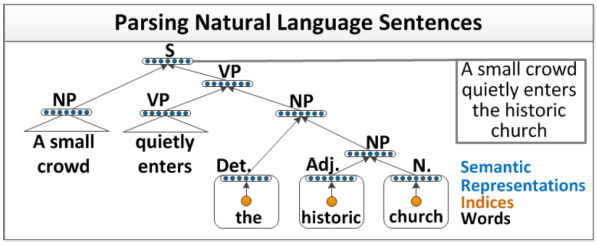

図 2.2: Recursive Neural Network（[20] より引用）

二値判定を行った．この論文内では，自己相互情報量 (Point-wise Mutual Infomation) を用いて，レビュー内の各フレーズに Sentiment Lexicon との共起のしやすさを計算し，そこから各レビューの極性を求めるという手法である．このことから，同一極性のある単語群は共起しやすいという仮定の基で，テキストのポジティブ・ネガティブの極性判定に約 75% の精度を出すことに成功している．また，Pang ら (2002) [17] は，感情分析をテキストのトピック分類問題の一種とみなし，三種類の分類アルゴリズム (Naive Bayes, Maximum Entropy, SVM) による分類精度を評価した．結果として，SVM がもっとも良い結果を出したことを示している．

また，対話のテキストは一般的にレビューに比べて語数が少ない傾向にあることから，Twitter [1] のツイートのようなテキストを研究データとして使用することが多い．ツイートの感情分析は Twitter Sentiment[2] や Tweetfeel[3] など既にウェブブラウザ上で動くアプリケーションが登場している．これは，ユーザがクエリを入力すると，そのクエリを含むツイートを検索し，そのツイートについて極性判定するようなシステムとなっている．このようなツイートの極性判定システムに関する研究として，Long (2011) ら [18] は，Sentiment Lexicon などのクエリ独立の特徴以外にも，文法などのクエリ依存の特徴を SVM に追加することで分類精度が上がることを示している．

2.2.2 ニューラルネットを用いた感情分析

近年では深層学習技術の発展により，自然言語処理の分野でも深層学習を用いた手法が提案されている．深層学習において，自然言語は Recurrent Neural Network (図 2.1) と Recursive Neural Network (図 2.2) が使われることが多い．両者は，どちらも文を単語分割して扱うが，Recurrent Neural Network が文の始めから単語を時系列的に入力するのに対し，Recursive Neural Network は文を木構造 (2 分木) で表現したものを葉ノードの単語から入力する違いがある．Socher ら (2011) [20] は Recursive Neural Network を用いて，

[1] https://twitter.com/
[2] http://twittersentiment.appspot.com/
[3] http://www.tweetfeel.com/

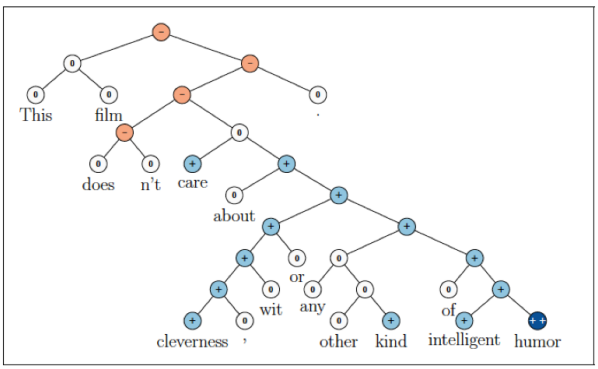

図 2.3: Sentiment Tree Bank の一例（[21] より引用）

木構造の文書の感情分析を行った．使用された木構造データは Sentiment Tree Bank [21] (図 2.3) と呼ばれ木構造データを用いた研究に広く利用されている．また，同様の技術を拡張したモデルを発表し，通常のモデルより精度が上がっていることを示している [22] [21] ．さらに，Tai ら (2015) は Recursive Neural Network の各ユニットを LSTM [23] に置き換えた Tree-LSTM モデルを提案している [24] ．同時期に，Zhu ら (2015) もツリー状に LSTM ユニットを繋いだモデル S-LSTM を提案している [25] ．

深層学習を用いた感情分析は, RNN に限らず, 畳み込みニューラルネットワーク (CNN) も使用されている．一般的に CNN は画像データを入力として使用するが，文書を入力として扱う場合，単語一つ一つをベクトル表現で表し，それを並べたもの (行列) を入力として扱うことによって文を学習させている．Kim (2014) [26] は CNN を用いた感情分析等の複数データの精度評価を行った．Santos ら (2014) [27] は，畳み込みニューラルネットワークを用いたツイートの感情分析を研究している．また，Zhang ら (2015) [28] は，CNN で使用するベクトル表現方法や畳み込み層，プーリング層などのパラメータを調整しながら実験を行い，評価している．近年，Twitter を用いた感情分析コンペティションも開催されている．2017 年には SemEval-2017 Task 4: Sentiment Analysis in Twitter [29] が開かれており，CNN と Bi-LSTM を用いたアンサンブル学習を使用したモデルが最も高い精度を示している [30]．また，ツイートのテキスト分類として，Dhingra [31] らはツイートのハッシュラベルを用いた新たなトピック分類手法 tweet2vec を提案し，既存のトピック分類である LDA よりも高精度の分類精度を実現している．

2.3 対話システムに関する研究

2.3.1 用例ベース対話システム

対話システムの研究については，音声・言語処理技術の高度化に伴い，対話技術への需要が高まっている．言語データから対話システムを構築する例として，用例ベース対話シ

ステムが考えられる [32] ．これは，発話と応答の組である用例を用いてシステムを構築するデータ駆動型の対話システムであり，このデータの量と品質，応答選択の精度が用例ベース対話システムの精度に大きく関わっている．このようなシステムにおいて，応答選択の精度をあげるために TF-IDF [33] や行列分解 [34] 等を用いたシステムを構築することで，より精度の高い対話システムを作成するアプローチがなされている．

2.3.2 Sequence-to-Sequence Model を用いたニューラル対話モデル

用例ベースのようなアプローチとは異なり，近年ではニューラルネットを用いた対話システムが研究されている．主に使用されるニューラルネットモデルとして，Sequence-to-Sequence (seq2seq, Encoder-Decoder) [1, 35] が挙げられる．これは入力を発話，出力を応答としたモデルであり，一つのシステムにより対話システムを完結させている．このモデルを使用した対話システムの例として，Vinyals [36] らはシンプルな seq2seq モデルを大規模対話データで学習することで対話システムを構築している．既存の対話システム CleverBot [4] と比較し，提案モデルの方がより精度の高い応答が為されていることが報告されている．また，Li ら [37] が提案した発話者の人格を考慮した対話文生成システムが挙げられる．このシステムでは seq2seq のエンコーダ部分の入力に前時刻の出力に加え，発話者の人格を現す特徴ベクトルを入力に加えることで，その人格に沿った一貫性のある出力が得られることを示している．Li らの研究を踏まえて，村上ら [38] はコメント配信時の日時などを考慮した数値予報マップから天気予報コメントを自動生成する手法を提案している．

seq2seq のフレームワークは対話の研究のみならず，機械翻訳の研究においても数多く用いられている．一例として，Wu ら [39] や Johnson [40] らは，アテンション機構 [41] [42] と双方向エンコーダ を適用した Encoder-Decoder を 8 層の多層 LSTM で構築している．8 層では勾配消失が発生するので Residual Connection を取り入れている．アテンション機構 は多層の Encoder の各層に対して行っている．また，普通の機械翻訳では対訳の 2 言語間で 1 つのモデルを学習するが，Johnson らはモデルのパラメータを共有して多言語間で 1 つのモデルを学習し，普遍的な翻訳知識を獲得している．seq2seq の拡張例として，Weng ら [43] は通常の seq2seq モデルに Word Prediction の機構を新たに加え，訓練時に目的の単語が出力されているかどうかを学習するモデルを提案し，既存の中国語－英語，ドイツ語－英語の機械翻訳システムよりも BLEU 値をそれぞれ 4.53，1.3 上昇させることに成功している．また，Eriguchi ら [44] は木構造を考慮したアテンション付きエンコーダデコーダを提案しており，既存のアテンション NMT よりも良い性能を示している．Luong ら [45] は seq2seq モデルにマルチタスク学習を適応し，文章の parsing と翻訳，事前学習を seq2seq モデルで行い，既存手法より精度が向上している．

[4] http://www.cleverbot.com/

2.3.3 Sequence-to-Sequence Model の問題点と解決に向けた取り組み

表 2.1: Emotional Chatting Machine による出力例（[46] より引用）

Post	Model		Response
A promise is what a cheater says to a fool.	Seq2Seq		A promise is what a cheater says to a fool.
	ECM	Like	Promise is a single-minded fool
		Happy	I'm just a happy fool
		Sad	I'm just a sad fool
		Disgust	A cheater says to a fool
		Angry	Whatever you believe or not, I trust in it!
Knowing the truth, I laughed myself to tears	Seq2Seq		After knowing the truth, I burst into tears.
	ECM	Like	Too lovely, it's too lovely.
		Happy	I can't help laugh myself to tears.
		Sad	After knowing the truth, I burst into tears.
		Disgust	Is this true?
		Angry	What do you mean? I don't understand.

seq2seq の問題点として，「汎用的で固定的な発話」が高頻度で出力されてしまう問題が存在する [2, 3] ．汎用的な対話とは，「そうですね」や「確かに」等の多くの状況下の返答として当てはまりうる発話を指し，特定のドメインに即した発話生成が困難である．固定的な対話とは，一つの発話には一つの応答しか返せないことを指し，例えば感情に応じた発話を生成することができない．また，ニューラルネットの学習精度は，学習データの量に依存しており，学習データが少量の場合文法が崩れやすい．従って，未だに企業向けのチャットシステムでは，ルールベースとの併用が多い [47] ．

上記のような問題に対し，情報量の多い発話を生成するモデルとして，Shao ら [48] は，seq2seq が長文生成と多様性に対応するために，Decoder の Self Attention と Segment Beam Search を適用しており，通常の seq2seq に比べ情報量の多い応答の生成に成功している．Yao ら [49] も，seq2seq の応答の情報量を増やすために，応答生成の際に入力発話の特徴を表すキーワードを入力情報として使用した対話システムを提案している．赤間ら [50] は seq2seq による応答生成モデルと転移学習を組み合わせた手法をスタイル制御に応用しており，通常の対話に比べ，キャラクタ性を考慮した応答を可能にしている．また，佐藤 [51] らは会話データに付随するタイムスタンプや発話内容のクラスタリングから獲得した季節や現在の話題といった発話状況を導入したニューラル対話モデルを提案し，応答選択テストによって提案手法の有効性を確認している．

また，対話システムにおける感情制御の論文として，Hasegawa ら [52] 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，悲しみ等の 9 つのカテゴリに分類した感情ラベル付き対話コーパスを構築し，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

定の感情を喚起させる応答の生成を試みている．Zhou ら [46] は seq2seq のモデルに感情を制御する機構を加えた Emotional Chatting Machine (ECM) を提案し，既存のシステムとは異なり感情の制御を可能とした．ECM は，ユーザがある発話とそれに応じた感情ラベルを ECM に入力することによって，入力した感情値に沿った応答を返すことを可能にした．ECM による実際の応答例を表 2.1 に示す．表から，通常の seq2seq では，一つの発話に対して単一の応答しか返答できないが，ECM は感情ラベルに応じた応答ができていることが分かる．現状では，上記のような手法を用いて発話内の多様性を広げる手法が検討されている．しかし，現状では会話ドメインと感情制御は個別に議論されているが，その二つを一度に考慮している研究は存在しない．従って，本報告では会話ドメインと感情制御を一度に考慮した手法を検討する．

3　対話システムの設計

本章では，本報告で提案した会話ドメインと感情制御の両方を考慮した対話システムについて述べる．対話システムの全体の概要を説明した後，使用した各手法について述べる．

3.1　システム概要

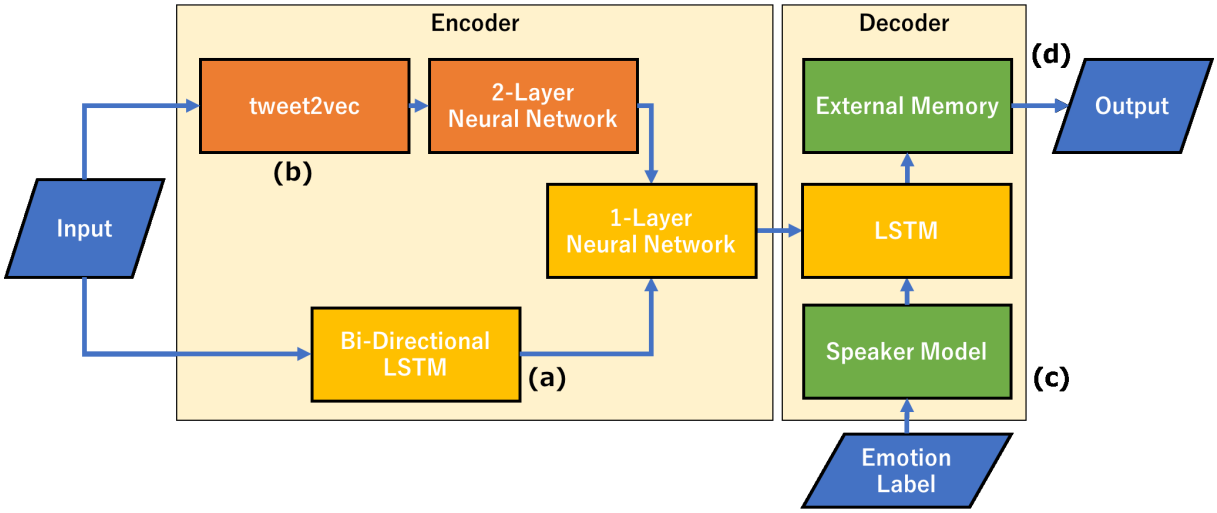

図 3.1: 提案モデルの概要図

本報告で提案する対話システムの概要について述べる．本提案システムの概要図を図 3.1 に示す．図 3.1 の各コンポーネントは機能別に色分けされており，seq2seq の大元であるエンコーダ部とデコーダ部は黄色，会話ドメインの推定に用いるコンポーネントは橙色で，感情制御を行うコンポーネントは緑色で示す．本システムでは，まず入力となる発話を Bi-Directional LSTM (図 3.1 (a)) と tweet2vec (図 3.1 (b)) に入力する．前者は入力をエンコーディングし，出力制御のデコーダの初期状態として渡すベクトルを生成する．後者は入力の会話ドメインの推定を行い，出力された推定結果ラベルを次層のニューラルネットに入力する．入力されたラベルは，二層の全結合層を通してラベルの分散ベクトルとして出力され，そのベクトルと Bi-Directional LSTM の出力を結合させたベクトルを一層の全結合層に通し，その出力を Decoder の LSTM に入力する．Decoder では，各時刻毎に単語が出力される．現在時刻に注目すると，前時刻に出力された単語と出力したい感情のラベルを Decoder に入力する（図 3.1 (c) ）．その後，Decoder で解析され，出力された状態ベクトルを External Memory (節 3.1 (d)) に入力し，確率値の最も高い単語を出力する．出力された単語は次状態の入力として使用され，文の最後を表すシーケンスが出力されるまで，単語の出力を繰り返す．

以上が，本提案システムの入出力の一連の流れである．また，本システムの具体的な学習方法については 3.5 節にて記述する．次節以降，本システムの根幹を為す seq2seq モデ

ルについて説明した後，システムを Encoder 側と Decoder 側の二つに分けて説明を行う．

3.2 Sequence-to-Sequence Model

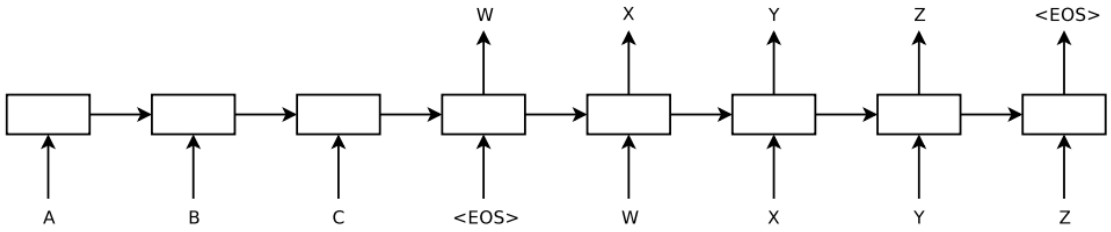

図 3.2: Sequence-to-Sequence Model（[1] より引用）

本提案手法では生成型対話システムとして，Sequence-to-Sequence [1] (seq2seq) を適用する．seq2seq は LSTM block (もしくは GRU block) を 二つ用いて，入力を処理するエンコーダと出力を生成するデコーダを組み合わせたモデルである (図 3.2) ．seq2seq の学習は対となるペア X, T によって行う．入力シーケンスを $X = \{x_1, x_2, \cdots, x_{|x|}\}$，デコーダの出力シーケンスを $Y = \{y_1, y_2, \cdots, y_{|y|}\}$，正解シーケンスを $Z = \{z_1, z_2, \cdots, z_{|z|}\}$ とする．seq2seq の順伝搬は，入力シーケンス $x_1, x_2, \cdots, x_{|x|}$ をエンコーダの LSTM block に順に入力する．入力シーケンスは言語を入力とする際は，単語の ID 列として表現される．エンコーダではその ID 列を単語分散表現ベクトルに変換して，LSTM へと入力する．エンコーダの LSTM block には出力層への出力はなく，LSTM block の cell state と hidden state を次ステップの LSTM block に渡すのみである．seq2seq のエンコーダ側の伝播式は以下のように記述される．

$$
\begin{aligned}
\mathbf{x}_t^{\text{embed}} &= \mathbf{W}_{\text{embedx}}\mathbf{x}_t, \\
\mathbf{c}_t, \mathbf{h}_t &= \mathbf{LSTM}(\mathbf{x}_t^{\text{embed}}, \mathbf{c}_{t-1}, \mathbf{h}_{t-1}).
\end{aligned}
$$

c, h は LSTM の cell state と hidden state である．$\mathbf{x}_t$ は x_t を表す one-hot ベクトルであり，$\mathbf{W}_{\text{embedx}}$ は入力の分散表現行列である．$\mathbf{x}_t^{\text{embed}}$ は x_t を表す分散表現ベクトルであり，LSTM に入力する単語ベクトルである．LSTM(·) は以下の伝播式で表される．

$$
\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_t^{\text{embed}} + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_t^{\text{embed}} + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_t^{\text{embed}} + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \\
\tilde{\mathbf{C}}_t &= \tanh(\mathbf{W}_C \mathbf{x}_t^{\text{embed}} + \mathbf{U}_C \mathbf{h}_{t-1} + \mathbf{b}_C), \\
\mathbf{C}_t &= \mathbf{f_t} \odot \mathbf{C}_{t-1} + \mathbf{i_t} \odot \tilde{\mathbf{C}}_t, \\
\mathbf{h}_t &= \mathbf{o_t} \odot \tanh(\mathbf{C}_t)
\end{aligned}
$$

ここで，$\sigma(\cdot)$ はシグモイド関数，$\odot$ は行列の要素積である．最後に $x_{|x|}$ を入力した時のエンコーダの cell state と hidden state をデコーダ側の初期状態として，デコーダの LSTM block に入力する操作が通常の seq2seq である．入力文は翻訳等のタスクの場合，入力単語を逆順序で入力する方が一般的に精度が上がると言われている．もしくは，入力文に対してエンコーダを二つ用意し，一つは順序通りに入力単語を解析するエンコーダ，もう一方は逆順序で入力単語を解析するエンコーダとする．各エンコーダの最後の hidden state を結合・集約した state をデコーダに渡す．以上の操作を数式で示すと以下の通りとなる．

$$
\begin{aligned}
\mathbf{c}_t^{bw}, \mathbf{h}_t^{bw} &= \mathbf{LSTM}(\mathbf{x}_t^{bw}, \mathbf{c}_{t-1}^{bw}, \mathbf{h}_{t-1}^{bw}), \\
\mathbf{h}^{\text{enc}} &= \sigma(\mathbf{W_s} \frac{1}{|x|} \sum_{t=0}^{|x|} [\mathbf{h}_t; \mathbf{h}_t^{bw}]),
\end{aligned}
\tag{3.1}
$$

式 3.1 は，逆順序入力の伝播式である．ここで，$[\cdot;\cdot]$ はベクトルの連結操作を表す．$\mathbf{x}_t^{bw}$ は逆順序入力の単語分散ベクトル，$\mathbf{c}_{t-1}^{bw}$ ，$\mathbf{h}_{t-1}^{bw}$ はそれぞれ逆順序入力に対する cell state と hidden state である．$\mathbf{h}_{\text{enc}}$ はエンコーダがデコーダに渡す最終状態であり，その状態は二つのエンコーダの状態を連結させたもの $\frac{1}{|x|}\sum_{t=0}^{|x|}[\mathbf{h}_t; \mathbf{h}_t^{bw}]$ を線形層 $\mathbf{W}_s$ と活性化関数 $\sigma(\cdot)$ に通した出力とする．

そして，デコーダの最初の LSTM block にデコード開始記号を入力し，その際の出力 y_t と正解 t_1 の誤差が損失となる．次ステップの LSTM block への入力は学習では z_1, 本番では y_1 となり，その時の出力 y_2 と対訳 z_2 の誤差が損失となる．これを一般化すると以下の式になる．

$$
\begin{aligned}
\mathbf{y}_t^{\text{embed}} &= \mathbf{W}_{\text{embedy}}\mathbf{y}_t, \\
\mathbf{z}_t^{\text{embed}} &= \mathbf{W}_{\text{embedy}}\mathbf{z}_t, \\
\mathbf{c}_t, \mathbf{h}_t &= \begin{cases} \mathbf{LSTM}(\mathbf{z}_t^{\text{embed}}, \mathbf{c}_{t-1}, \mathbf{h}_{t-1}) & \text{(訓練時)} \\ \mathbf{LSTM}(\mathbf{y}_t^{\text{embed}}, \mathbf{c}_{t-1}, \mathbf{h}_{t-1}) & \text{(学習時)} \end{cases}
\end{aligned}
$$

$\mathbf{y}_t^{\text{embed}}$，$\mathbf{z}_t^{\text{embed}}$ はそれぞれ y_t，z_t を表す単語分散ベクトルでありこれを繰り返して y_{n+1} と z_{n+1} の損失まで累積する．この累積損失を誤差逆伝搬してパラメータ更新を行う．デコーダ部では，以下の式で連続的にソフトマックス関数を用いて出力単語を予測する．

$$
\begin{aligned}
p(Y|X) &= \prod_{t=1}^{|y|} p(y_t|x_1, x_2, \cdots, x_{|x|}, y_1, y_2, \cdots, y_{t-1}), \\
&= \prod_{t=1}^{|y|} \frac{\exp(f(\mathbf{h}_{t-1}, \mathbf{e}_{y_t}))}{\sum_{y'} \exp(f(\mathbf{h}_{t-1}, \mathbf{e}_{y'}))}
\end{aligned}
$$

$\mathbf{e}_{y_t}$ は時刻 t での出力ステートを単語分散ベクトルに変換したものである．$f(\mathbf{h}_{t-1}, \mathbf{e}_{y_t})$ は $\mathbf{h}_{t-1}$ と $\mathbf{e}_{y_t}$ 間の活性化関数である．

3.3 Encoder の設計

本章では，本提案手法における Encoder 側の設計について説明する．まず，本システムで発話のドメイン推定として用いた tweet2vec について説明した後, 本システムの Encoder 側の順伝播について述べる．

3.3.1 会話ドメインの推定

本手法では，会話ドメインの推定手法として tweet2vec [31] を適用する．tweet2vec はツイートなどの短文を分類する手法となっており，具体的には，入力文を一文字毎に GRU [53] ユニットを用いた双方向エンコーダ (Bi-GRU Encoder) に入力し，得られた入力文の Embedding をソフトマックス層に通して，テキスト分類を行う手法である．tweet2vec は文字を単位として入力文をベクトル化するため，従来の LSI や LDA 等のトピック分類手法に比べ，疎性の強い短文のテキスト分類でも効果を期待できる．対話文はニュース記事とは異なり，長文になることが非常に稀であるので tweet2vec は対話文のトピックを推定するには適した手法であると考えられる．

tweet2vec の具体的な計算方法について，以下に述べる．入力文を文字単位で分解したものを $c_1, c_2, \cdots, c_m$ とする．それぞれの文字は，$|C|$ 次元の one-hot ベクトルとして表

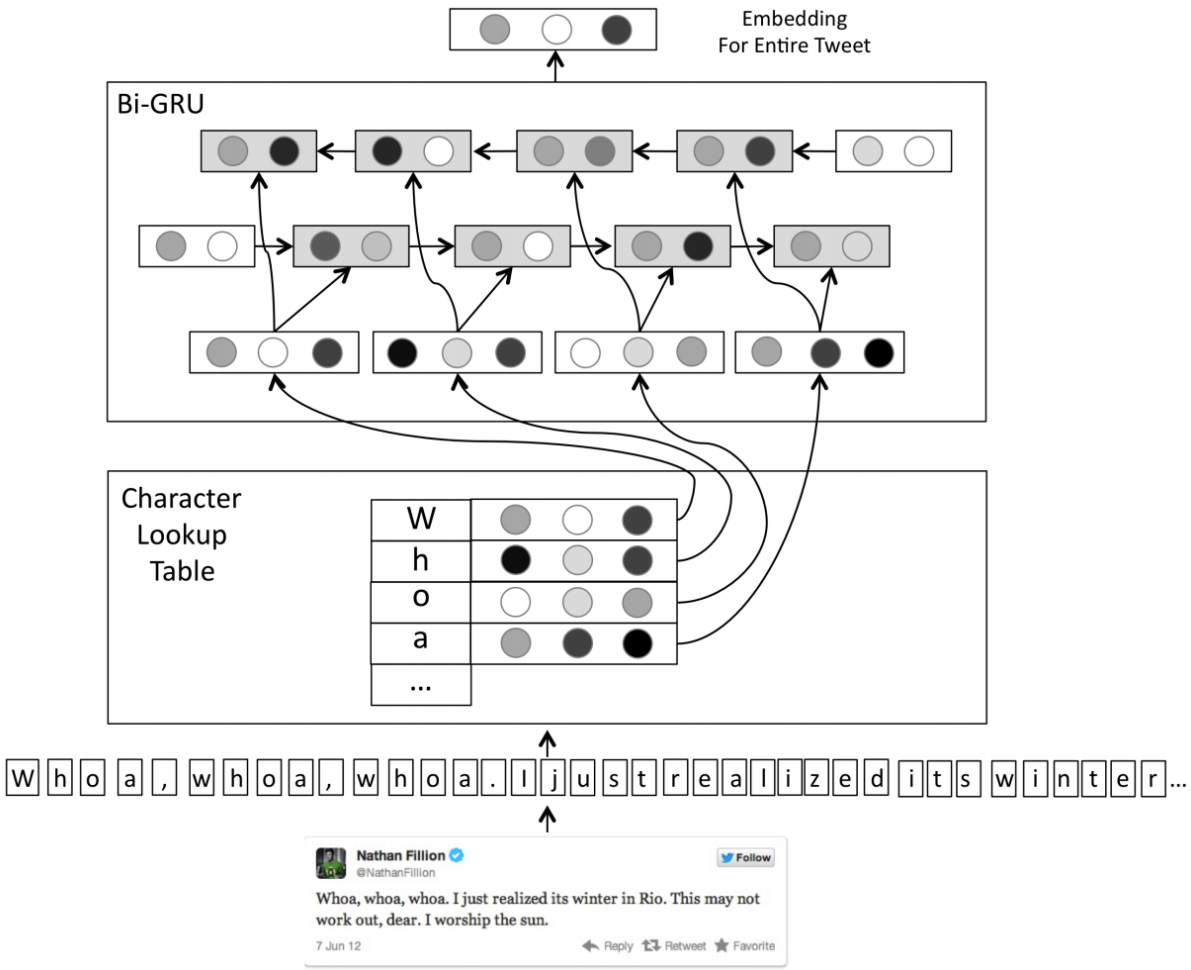

図 3.3: Tweet2vec: Bi-GRU Encoder 部モデル図（[31] より引用）

現される．$|C|$ は文字として認識されうるユニコードキャラクタ集合の要素数である．これらの one-hot ベクトルは行列 $P_C \in \mathbb{R}^{|C| \times d_c}$ によって，分散表現ベクトル空間に射影される．d_c は 1 キャラクタに与えられるベクトルの次元数である．入力する文字の分散表現ベクトルを $x_1, x_2, \cdots, x_m$ (次元数：d_c) とすると，Bi-GRU Encoder (図 3.3) は初期状態 h_0 から始まり，入力ベクトルに対しシーケンシャルに $h_1, h_2, \cdots, h_m$ を計算する．したがって，Bi-GRU Encoder は以下の式で表される．

$$
\begin{aligned}
\mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r h_{t-1} + \mathbf{b}_r), \\
\mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z h_{t-1} + \mathbf{b}_z), \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_t \mathbf{x}_t + \mathbf{U}_h (r_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t, \\
\mathbf{e}_t &= \mathbf{W}^f \mathbf{h}_m^f + \mathbf{W}^b \mathbf{h}_0^b.
\end{aligned}
$$

$\odot$ はベクトルの要素積を表す．$\mathbf{r}_t$，$\mathbf{z}_t$ はリセットステートとアップデートステートと呼ばれる．$\tilde{\mathbf{h}}_t$ は実際の出力である $\mathbf{h}_t$ から計算された出力ステート候補である．$\mathbf{W}_r$，$\mathbf{W}_z$，$\mathbf{W}_h$ は $d_h \times d_c$ 次元の行列である．$\mathbf{U}_r$，$\mathbf{U}_z$，$\mathbf{U}_h$ は $d_h \times d_h$ 次元の行列である (d_h は隠れ層の次元数)．$\mathbf{h}_m^f$ は forward-GRU の最終ステートであり，$\mathbf{h}_0^b$ は backward-GRU の最終ステートである．得られた二つの最終ステートから，$d_t \times d_h$ 次元の行列である $\mathbf{W}^f$，$\mathbf{W}^b$ と，次元数 $d_t \times 1$ のバイアス項 $\mathbf{b}$ を用いて，最終的な入力文のベクトル $\mathbf{e}_t$ が得られる．

得られたベクトルは出力が分類ラベル数 L である一層の線形結合層を通る．具体的には下記のソフトマックス関数を用いてトピックの確率を計算する．

$$P(y = j|e) = \frac{\exp(\mathbf{w}_j^T \mathbf{e} + \mathbf{b}_j)}{\sum_{i=1}^{L} \mathbf{w}_i^T \mathbf{e} + \mathbf{b}_j}.$$

目的関数として，Categorical Cross Entropy Loss [54] を用いて予想ラベルと実際のラベルの誤差を計算する．式は以下の通りである．

$$J = \frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{L} -t_{i,j} \log(p_{i,j}) + \lambda \| \Theta \|^2 \tag{3.2}$$

B はバッチサイズ，L は予測するラベル（クラス）数，$p_{i,j}$ は i 番目の文が j 番目のラベルである確率値である．また，$t_{i,j} \in \{0, 1\}$ i 番目の文が j 番目のラベルであるかどうかの二値である．L2 正則化項の重みとして，λ を使用している．

3.3.2 Encoder の順伝播

本節では，前節で述べた Encoder と Decoder の設計を踏まえて，入力が与えられた時の提案手法の伝播式を記述する．

図 3.4 に提案手法における Encoder のニューラルネットワークの構成図を示す．図 3.4 の $\otimes$ はベクトルの連結，$\oplus$ はベクトルの和を示す．本システムでは，まず入力となる発話を Bi-Directional LSTM と tweet2vec に入力する．前者は入力をエンコーディングし，出力制御のデコーダの初期状態として渡すベクトルを生成する．後者は入力の会話ドメインの推定を行い，出力された推定結果ラベルを次層のニューラルネットに入力する．対話文対の内，入力シーケンスを $X = \{x_1, x_2, \cdots, x_{|x|}\}$，対応する出力シーケンスを $Y = \{y_1, y_2, \cdots, y_{|y|}\}$ とする．$|x|$, $|y|$ はそれぞれ X, Y のサイズである．tweet2vec の伝播式は以下の様に示される．

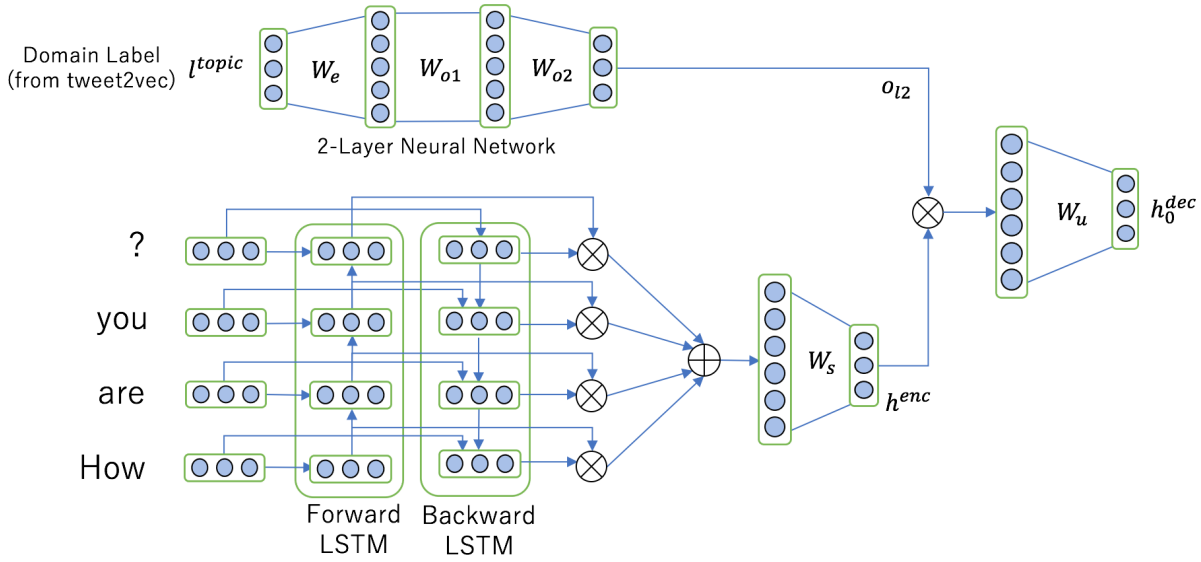

図 3.4: 提案手法における Encoder のニューラルネットワーク構成図

$$
\begin{aligned}
\mathbf{h}_t^{fw} &= \mathbf{GRU}(\mathbf{h}_{t-1}^{fw}, \mathbf{x}_t^{fw}), \\
\mathbf{h}_t^{bw} &= \mathbf{GRU}(\mathbf{h}_{t-1}^{bw}, \mathbf{x}_t^{bw}), \\
\mathbf{e}^{\text{tweet}} &= \mathbf{W}^{fw}\mathbf{h}_m^{fw} + \mathbf{W}^{bw}\mathbf{h}_0^{bw}, \\
l^{\text{tweet}} \sim \mathbf{o}^{\text{tweet}} &= \text{softmax}(\mathbf{W}_t\mathbf{e}^{\text{tweet}}).
\end{aligned}
$$

$\mathbf{x}_t^{fw}$ は正順序入力，$\mathbf{x}_t^{bw}$ は逆順序入力である．$\mathbf{h}_m^{fw}$ は forward-GRU の最終ステートであり，h_0^{bw} は backward-GRU の最終ステートである．得られた二つの最終ステートから，$d_t \times d_h$ 次元の行列である $\mathbf{W}^{fw}$，$\mathbf{W}^{bw}$ から最終的な入力文のベクトル $\mathbf{e}^{\text{tweet}}$ が得られる．d_h は隠れ層の次元数であり，d_t は $\mathbf{e}^{\text{tweet}}$ の次元数である．$\mathbf{o}^{\text{tweet}}$ は各トピックラベルの確率分布であり，実際の正解ラベル l^{tweet} との誤差により学習を行う．

予測されたラベルは，二層ニューラルネットを通してラベルの分散ベクトルとして出力され，そのベクトルと式 (3.4) の Bi-Directional LSTM の出力を結合させたベクトルを Decoder 層に入力する．Bi-Directional LSTM の伝播式は以下のように記述される．

$$
\begin{aligned}
\mathbf{c}_t^{fw}, \mathbf{h}_t^{fw} &= \mathbf{LSTM}(\mathbf{x}_t^{fw}, \mathbf{c}_{t-1}^{fw}, \mathbf{h}_{t-1}^{fw}), \\
\mathbf{c}_t^{bw}, \mathbf{h}_t^{bw} &= \mathbf{LSTM}(\mathbf{x}_t^{bw}, \mathbf{c}_{t-1}^{bw}, \mathbf{h}_{t-1}^{bw}), \qquad (3.3) \\
\mathbf{h}^{\text{enc}} &= \sigma(\mathbf{W}_s \frac{1}{|x|} \sum_{t=0}^{|x|} [\mathbf{h}_t^{fw}; \mathbf{h}_t^{bw}]). \qquad (3.4)
\end{aligned}
$$

$\mathbf{c}^{fw}$，$\mathbf{h}^{fw}$ は順方向 LSTM の cell state と hidden state である．$\mathbf{c}_t^{bw}$ ，$\mathbf{h}_t^{bw}$ はそれぞれ逆順序入力用 LSTM の cell state と hidden state である．

次に，推定されたドメイン情報と入力をエンコーディングした情報を合わせるために，tweet2vec から推定された会話ドメインラベルを表す one-hot ベクトル $\mathbf{l}^{\text{topic}}$ （ $\mathbf{o}^{\text{tweet}}$ の確率分布において最も値の高いドメインを指す）を用いて，会話ドメインの情報を Bi-Directional LSTM の出力である $\mathbf{h}^{\text{enc}}$ に加える．具体的には，ドメインラベルは二層のニューラルネットを通じてベクトルとなり，そのベクトルと $\mathbf{h}^{\text{enc}}$ を連結させたベクトルを一層のニューラルネットに入力する．伝播式は以下の式で示される．

$$
\begin{aligned}
\mathbf{e}_l &= \mathbf{W}_{\mathbf{e}}(\mathbf{l}^{\text{topic}}), \\
\mathbf{o}_{l1} &= \text{relu}(\mathbf{W}_{l1}\mathbf{e}_l), \\
\mathbf{o}_{l2} &= \text{relu}(\mathbf{W}_{l2}\mathbf{o}_{l1}), \\
\mathbf{h}_0^{\text{dec}} &= \mathbf{W}_{\mathbf{u}}[\mathbf{h}^{\text{enc}}; \mathbf{o}_{l2}].
\end{aligned}
$$

デコーダ部の LSTM の隠れ層の初期値 $\mathbf{h}_0^{\text{dec}}$ を与えた後，デコード開始を示す文頭記号 (<BOS>) を入力することで，デコーダは出力を開始する．

3.4 Decoder の設計

本提案手法では, seq2seq モデルの Decoder に, Speaker Model [37] と External Memory [46] を使用する．Speaker Model は Decoder の入力部にシステムが出力に持たせたい感情を制御するための感情ラベルを入力するために使用する．また，その感情ラベルの情報を加味した LSTM の出力を External Memory に入力することで感情語彙の出力を制御する．本章では，Speaker Model と External Memory について説明する．Decoder の具体的な設計は 3.4.3 節に記述する．

3.4.1 Speaker Model

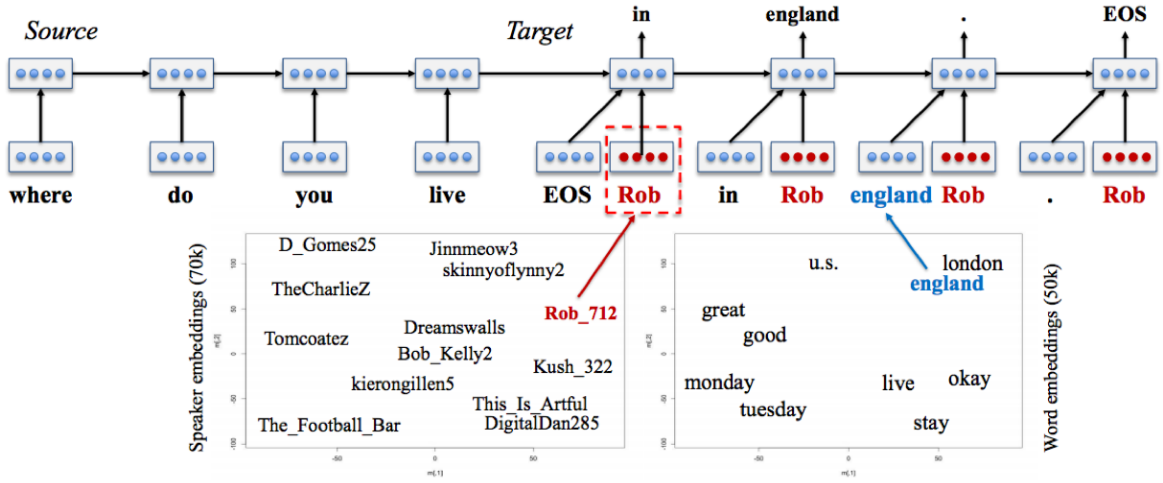

図 3.5: Speaker Model（[37] より引用）

本報告では，デコードの入力部分に対して感情ラベルの入力を行う．ラベルの入力手法は，Li ら [37] が提案した Speaker Model (図 3.5) を適用する．Speaker Model では，デコーダの入力に単語ベクトルだけでなく，外的情報を表したベクトルを同時に入力している．Speaker Model のデコーダ側の伝播式は以下のように記述される．

$$
\begin{aligned}
\mathbf{y}_t &= [\mathbf{e}_y; \mathbf{e}_{\text{label}}], \\
\mathbf{c}_t, \mathbf{h}_t &= \mathbf{LSTM}(\mathbf{y}_t, \mathbf{c}_{t-1}, \mathbf{h}_{t-1}).
\end{aligned}
$$

ここで，$[\cdot;\cdot]$ はベクトルの連結操作を表し，e_y は単語の分散ベクトル（通常の seq2seq の入力ベクトル），e_{label} は外的情報を表現した分散ベクトル（本提案手法では感情ラベルとして用いる）である．$\mathbf{y}_t$ は $\mathbf{e}_y$ と $\mathbf{e}_{\text{label}}$ を結合させたベクトルであり，このベクトルを通常通り seq2seq モデルのデコーダ部 LSTM block の入力とする．このように，デコーダの隠れ状態にメタ情報を入力することで，そのメタ情報が考慮された出力の生成が可能となる．本提案手法では感情ラベルをデコーダに入力することで，モデルが感情ラベルに応じた発話ができるように制御する．感情ラベルの具体的な説明は 5 章に記述する．

3.4.2 External Memory

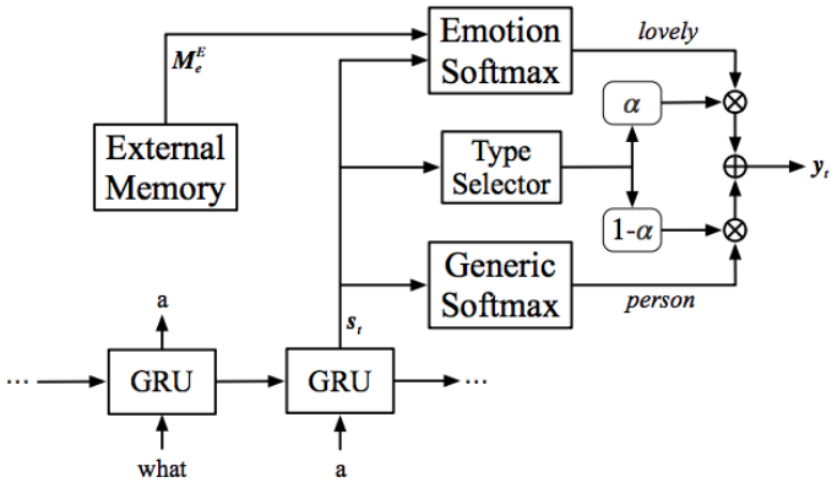

図 3.6: External Memory（[46] より引用）

また，本報告では感情の制御のために Decoder に External Memory を適用する．External Memory は Emotional Chatting Machine [46] で提案された機構であり，デコード時の出力に感情語彙用のメモリを用意し，システム側で感情語彙を出力するか一般語彙を出力するかを選択することが可能な手法である．Speaker Model は LSTM の状態遷移の入力として使用することで間接的に出力に感情を考慮させることが可能であるが，出力単語を直接的に制御することは難しい．感情語彙は他の一般語彙とは異なり，強い感情を発現させる要因になりやすいことが報告されている [55] ．External Memory はデコーダの出力部に直接作用することで，感情語彙の発現を操作しやすく，より直接的に感情のコントロールに干渉できる．従って，External Memory を用いて感情語彙の制御を行うことで， Speaker Model 単体よりも直接的に感情の制御が期待できる．External Memory の伝播式は以下の通りである．

$$
\begin{aligned}
\alpha_t &= \sigma(\mathbf{v}_u^T \mathbf{s}_t), \\
P_g(y_t = w_g) &= \mathrm{softmax}(\mathbf{W}_g^o \mathbf{s}_t), \\
P_e(y_t = w_e) &= \mathrm{softmax}(\mathbf{W}_e^o \mathbf{s}_t), \\
y_t \sim \mathbf{o_t} = P(y_t) &= \begin{cases} (1-\alpha_t) P_g(y_t = w_g), \\ \alpha_t P_e(y_t = w_e). \end{cases}
\end{aligned}
$$

ここで，$\sigma(\cdot)$ はシグモイド関数，$\alpha_t \in [0, 1]$ は感情語彙である w_e と一般語彙である w_g の選択を調整する実数値である．P_g と P_e はそれぞれ一般語彙群と感情語彙群の確率分

布である．$P(y_t)$ は最後に求められる全単語の確率分布となる．P_g と P_e の二つの確率分布は重複する単語を有しておらず，$P(y_t)$ はこの二つの確率分布を結合した分布であることに注意したい．

本提案手法の損失関数は，出力単語 $\mathbf{o_t}$ と訓練コーパス内の正解分布 $\mathbf{p_t}$ 間のクロスエントロピー誤差を使用する．加えて，正則化項として External Memory に対応した項を損失関数に組み込む．つまり，感情語彙か一般語彙の選択を制約する．入力シーケンスを $X = \{x_1, x_2, \cdots, x_{|x|}\}$，Decoder の出力シーケンスを $Y = \{y_1, y_2, \cdots, y_{|y|}\}$ とすると，損失関数 $L(\theta)$ は以下の式で記述される．

$$L(\theta) = -\sum_{t=1}^{|y|} \mathbf{p_t} \log(\mathbf{o_t}) - \sum_{t=1}^{|y|} q_t \log(\alpha_t) \tag{3.5}$$

$q_t \in 0, 1$ は Y における感情語彙か一般語彙かの選択が正解であるかどうかを指す．つまり，選択が正しい場合 $q_t = 1$ ，誤っている場合 $q_t = 0$ となる．上記の損失関数の値が最小になるように，学習を進める．

本実験では，事前にコーパス内の単語から感情語彙を選択する．感情語彙の具体的な選択手法は 5 章に記述する．

3.4.3 Decoder の順伝播

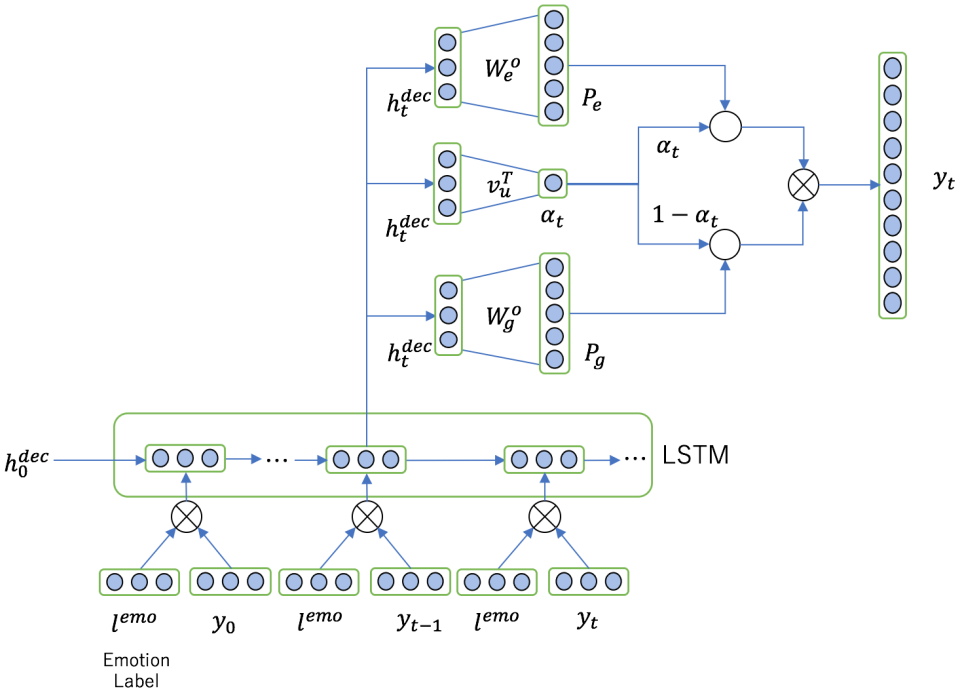

図 3.7: 提案手法における Decoder のニューラルネットワーク構成図

Decoder のニューラルネットワークの構成図を図 3.7 に示す．図 3.7 での ∘ はベクトルとスカラ値の積算を意味する．Decoder では，各時刻毎に前時刻に出力された単語と出力したい感情のラベル $\mathbf{l}^{\mathrm{emo}}$ を Decoder に入力する（式 (3.6)）．その後，Decoder で解析され，出力された状態ベクトルを External Memory [46] に入力し，確率値の最も高い単語を出力する．出力された単語は次状態の入力として使用され，文の最後を表すシーケンスが出力されるまで，単語が出力される．伝播式は以下の様に表される．

$$
\begin{aligned}
\mathbf{c}_t^{\mathrm{dec}}, \mathbf{h}_t^{\mathrm{dec}} &= \mathbf{LSTM}([\mathbf{y}_{t-1}; \mathbf{l}^{\mathrm{emo}}], \mathbf{c}_{t-1}^{\mathrm{dec}}, \mathbf{h}_{t-1}^{\mathrm{dec}}), \\
\alpha_t &= \sigma(\mathbf{v}_u^T \mathbf{s}_t), \\
P_g(y_t = w_g) &= \mathrm{softmax}(\mathbf{W}_g^o \mathbf{s}_t), \\
P_e(y_t = w_e) &= \mathrm{softmax}(\mathbf{W}_e^o \mathbf{s}_t), \\
\mathbf{y}_t = P(y_t) &= \begin{cases} (1 - \alpha_t) P_g(y_t = w_g), \\ \alpha_t P_e(y_t = w_e). \end{cases}
\end{aligned}
\tag{3.6}
$$

3.5 提案手法の学習方法

本節では，本提案システムの具体的な学習方法について述べる．現状，ドメインに特化した対話データや感情ラベルが付随した対話データは収集することが困難である．従って，本システムの学習は，より対話の精度を向上させるために大量の一般対話対データで Pre-training を行い（図 3.8），その重みを使用して特定ドメイン対話対データで Fine Tuning（図 3.9）を行う．また，学習モデルの各次元数などの具体的なパラメータの数値については 5.2 節で示す．

3.5.1 Pre-training

まず初めに，Pre-training を図 3.8 のように seq2seq 及び External Memory と tweet2vec に分けて別々に学習する．seq2seq 及び External Memory は学習データとして，5.1 節で収集する一般ドメイン対話コーパスを用いる．大量データとして一般ドメイン対話を事前学習することで，事前学習しない場合に比べて任意の発話に対してより堅牢に応答を生成できると考えられる．Pre-training の段階で，External Memory において感情語彙の分散表現が学習される．2018 年現在，感情ラベルが付随した対話データは存在しないため，本報告では収集した対話データ全てに擬似的に感情ラベルを付与する．ラベルの付与方法として，Decoder の感情ラベルの入力は訓練データの出力文内の感情語彙の個数で判定を行う．具体的には，ポジティブもしくはネガティブの感情語彙の個数が等しくない場合，感情語彙が多い方を出力文の感情として感情ラベルを指定する．ポジティブの感情語彙とネガティブの感情語彙の個数が等しい場合，出力文の感情はニュートラルとして感情

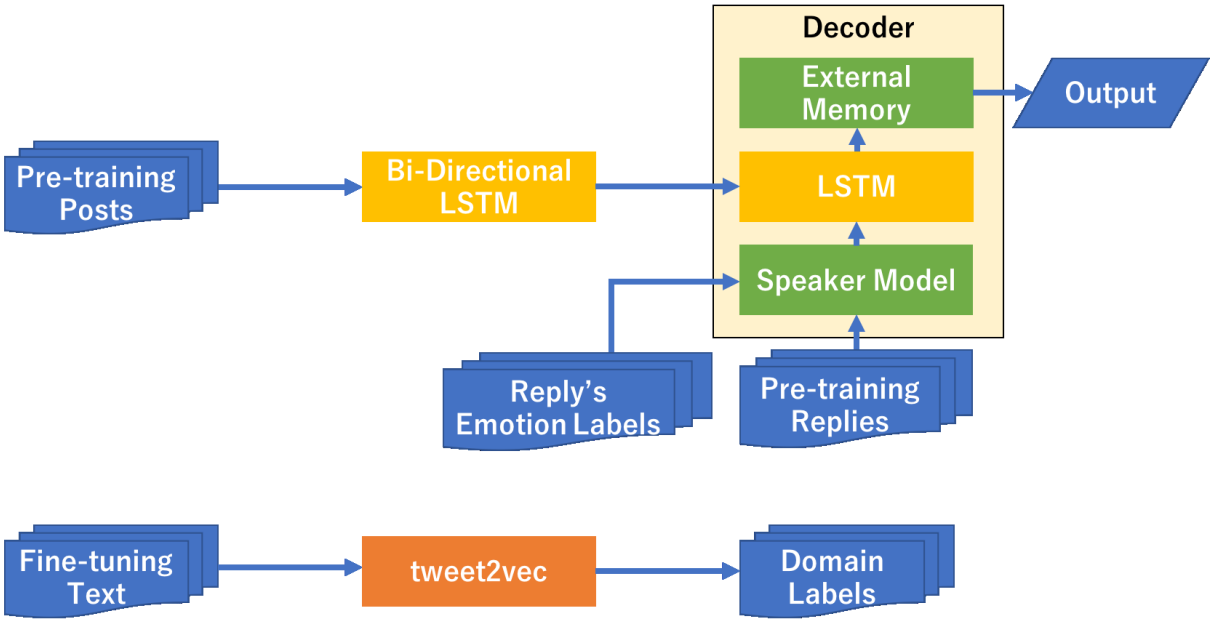

図 3.8: Pre-training 時の提案モデルの学習概要図：Pre-training 時は会話ドメインの推定部の学習と感情制御・対話生成部分の学習を分けて実行する．

ラベルを指定する．

seq2seq では学習コーパスの語彙数がモデルの計算量に直結するため，コーパスの作成時に語彙数が設定した上限単語数を超えた場合，低頻度語を事前に学習した word2vec を用いて単語の置き換えを行う．word2vec は学習コーパスとして，日本語 wikipedia のダンプデータ [1] を用いて学習を行う．低頻度語が word2vec 内に含まれる場合，word2vec 内の分散表現を用いてコサイン類似度が高い 50 件を抽出する．抽出した単語リストを上位から観測し，上限単語数のサイズの単語辞書に含まれている単語である場合，その単語と低頻度語を置き換える．リスト内に，word2vec に置き換え候補である低頻度語のデータが存在しない，もしくはコーパス内に単語が存在しない場合は unknown tag (<unk>) で置き換える．学習終了後に各エポック毎のロスを計算し，バリデーションセットのロスが低いモデルを Fine Tuning 時の初期値に設定する．ロス関数は式 (3.5) を用いてロスの計算を行う．

tweet2vec は 5.1 節で収集する特定会話ドメイン対話コーパスを用いて学習を行う．ロス関数は式 (3.2) を用いてロスの計算を行う．

3.5.2 Fine Tuning

Pre-training の終了後，システムの Fine Tuning（図 3.9）を行う．Fine Tuning を行うことによって，システムがトピックに沿った対話を行うことができる．Fine Tuning では，

[1] http://dumps.wikimedia.org/jawiki/latest/jawiki-latest-article.xml

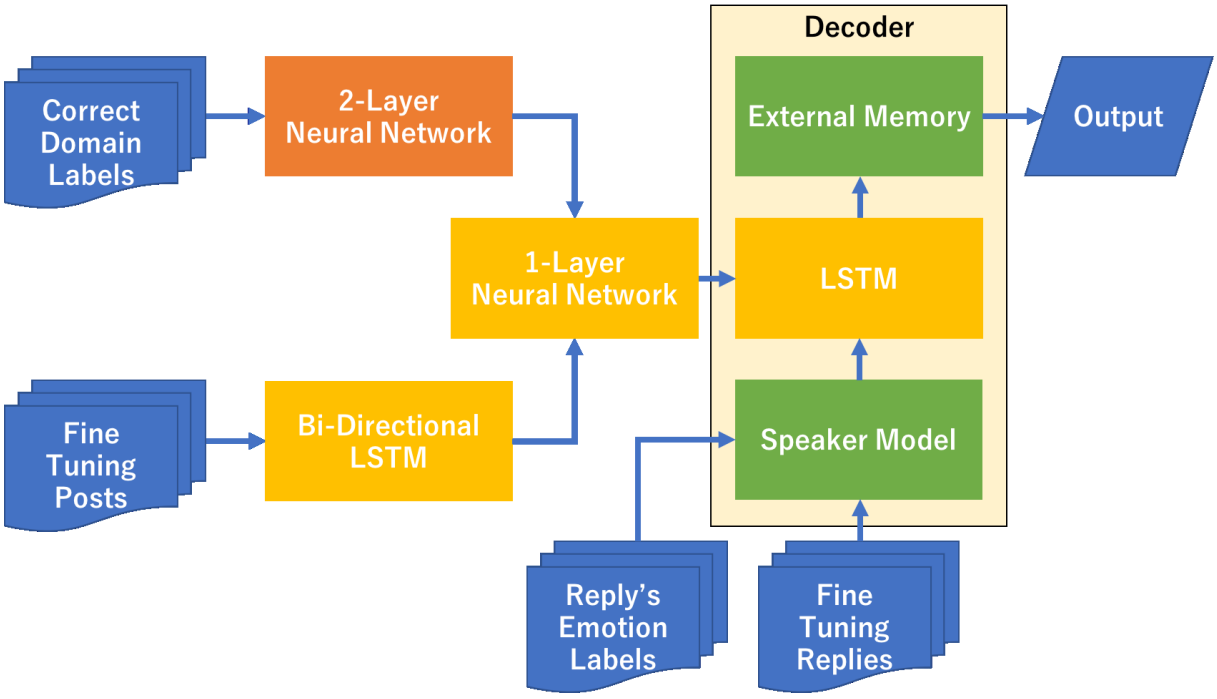

図 3.9: Fine Tuning 時の提案モデルの学習概要図

5.1 節で収集した特定会話ドメイン対話コーパスを用いて，学習を行う．Fine Tuning 時は，Decoder に渡す際に入力であるラベルの分散表現を学習するため，入力の発話と正解の会話ドメインラベルを与えて学習を行う．ラベルの分散表現と入力発話がエンコードされたベクトルの線形結合を Decoder に渡して出力と正解データとの誤差を逆伝搬してモデルを学習する．ロス関数は Pre-training 時と同様 式 (3.5) を用いてロスの計算を行う．感情ラベルの決定は Pre-training 時と同様の手法を用いる．低頻度語も Pre-training 時と同様に，word2vec を用いて置き換えを行った辞書を使用する．

4　評価指標

本章では，本報告で提案したシステムの評価に使用した指標について述べる．本提案手法の評価方法として，人手評価と自動評価の二つの方法で評価を行う．以下に各評価方法について述べる．

4.1　人手評価

事前に用意したテスト文に対し，手法に対する事前知識のない 5 名に評価してもらい，各評価値の平均を本実験の評価値とする．人手評価を自動評価とは別に行う理由として，対話システムの自動評価指標は 2018 年現在確立されている手法が少なく，評価の信頼性に欠けるためである．従って，本実験では人手評価をメインの評価に据えて実験を行う．本実験では，提案手法と既存の seq2seq との比較を Task1 で行い，提案手法の感情制御に関する評価を Task2 として行なっている．既存手法と提案手法の出力はアノテートの公平性のため，どちらの出力であるかは伏せた状態でのアノテートを行う．評価する指標を以下に述べる．

4.1.1　Task1: 提案手法 と seq2seq の比較

Task1 では提案手法と既存手法の出力について比較し，以下の 4 つの評価指標により評価を行う．また，各システムの出力はランダムシャッフルされており，どちらが提案手法か既存手法であるかは評価者からはわからない前提で評価を行う．

- **流暢性** (Fluency)：リプライが文として自然で文法的に大きな欠陥がないかを評価する．ここでは，文意が壊れない小さな欠陥は流暢性ありとして判断する．評価値は “0”（流暢性なし） もしくは “1”（流暢性あり） で判定され，精度はマクロ平均によって算出する．具体的には，以下の式により流暢性の精度を決定する．

$$\text{Fluency} = \frac{1}{T}\sum_{t=0}^{T} \text{fluency}_t$$

  T は全テスト文数，$\text{fluency}_t \in \{0, 1\}$ は t 番目の出力文に対して評価者が評価した評価値である．既存手法，提案手法それぞれで Fluency を算出し，評価者に対するマクロ平均の値を比較することで，流暢性を評価する．

- **一貫性** (Consistency)：入力発話に対して整合性が取れている応答を生成できたかどうかを評価する．流暢性が無いものについては一貫性も無いものとす

る．一貫性の精度は流暢性の精度と同様，評価者に対するマクロ平均によって算出する．式を以下に示す．

$$\text{Consistency} = \frac{1}{T}\sum_{t=0}^{T} \text{consistency}_t$$

$\text{consistency}_t \in \{0, 1\}$ は t 番目の出力文に対して評価者が評価した評価値である．評価値は “0”（一貫性なし） もしくは “1”（一貫性あり） で判定される．既存手法，提案手法それぞれで Consistency を算出し，評価者に対するマクロ平均の値を比較することで，一貫性を評価する．

- **会話ドメイン整合性** (`Domain Consistency`)：入力発話とその会話ドメインが与えられており，さらに既存手法と提案手法の二つの応答が与えられた時に会話ドメインがより考慮されている発話を選択する．評価の公平性のため，評価者はどちらが既存手法か提案手法は事前に知らされない．会話ドメイン整合性の評価値算出は以下の式で記述される．

$$\text{DomainConsistency} = \frac{1}{T}\sum_{t=0}^{T} \text{choice}_t$$

$\text{choice}_t \in \{0, 1\}$ は t 番目のシステムを選んだかどうかの値である．選択した場合 1 , 選択しなかった場合 0 となる．提案手法と既存手法の DomainConsistency について，評価者に対するマクロ平均を評価指標とし，それを合計すると 1.0 となる．

- **感情の豊かさ** (`Sentiment`)：入力発話とその会話ドメインが与えられた元で，既存手法と提案手法の二つの応答を比較して感情がより発現している発話を選択する．評価の公平性のため，評価者はどちらが既存手法か提案手法かは事前に知らされない．感情の豊かさの評価値算出は以下の式で記述される．

$$\text{Sentiment} = \frac{1}{T}\sum_{t=0}^{T} \text{choice}_t$$

$\text{choice}_t \in \{0, 1\}$ は t 番目のシステムを選んだかどうかの値である．選択した場合 $\text{choice} = 1$ ，選択しなかった場合 $\text{choice} = 0$ となる．提案手法と既存手法の Sentiment について，評価者に対するマクロ平均を評価指標とし，それらを合計すると 1.0 となる．

流暢性と一貫性は，出力文自体の評価であり，既存の seq2seq と比べて応答文が崩れていないかを判定するために利用する．会話ドメイン整合性と感情の豊かさは提案手法の特性を検証するために利用する．

4.1.2 Task2: 提案手法単体の評価

次に，Task2 の評価手法について述べる．Task2 は提案手法単体の評価であり，感情語彙の制御に対する評価を行う．以下に指標の説明を記述する．

- **感情の出現** (Emotion Tag)：感情ダグに対し，出力文に感情が発露しているかどうか評価する．4.1.1 節と同じ評価者に，提案手法が生成した応答について，ポジティブ，ネガティブ，ニュートラルかを判定してもらった．この時のテストセットは Task1 とは異なるデータを用いる．判定が難しい場合は判別不能とした．このとき，感情語彙の有無については考慮せず，あくまで文全体の感情を推定し，評価するよう指示した．感情制御の評価は以下の数式によって行う．

$$\mathrm{EmotionTag} = \frac{1}{T}\sum_{t=0}^{T}\mathrm{correct}_t$$

  $\mathrm{correct}_t \in \{0, 1\}$ は t 番目の出力文に対し，適切な感情値をつけたかどうかを示す値である．正解感情ラベルと同じ感情値をラベリングした場合 1，それ以外の場合は 0 とする．また，文構造の崩れ等で判別不能のラベルが付けられたテキストに関しては評価から除外する．評価者に対するマクロ平均を評価値とする．

Task2 の感情制御は感情語彙による感情制御の精度を調査するための指標である．提案手法の感情語彙の出力の制御自体の評価は 4.2.4 章の指標で行う．

4.2 自動評価

ここからはシステム精度を測る自動評価指標について述べる．

4.2.1 Tweet2vec の精度評価

tweet2vec が入力に対して，正確に会話ドメインの特定ができているかを調査する．具体的には，テスト文 300 件に対して以下の式で与えられる accuracy を測定する．

$$\text{Accuracy} = \frac{\text{会話ドメイン推定成功数}}{\text{テスト文の総数}} \tag{4.1}$$

本実験では，データ数の制限と実験の簡単化のため会話ドメイン数は二つに制限している．

4.2.2 BLEU

機械翻訳等のシステムにおける自動評価法の中で最も代表的な手法が BLEU [56] である．BLEU 値が高い程精度が良いシステムである．BLEU は以下の式を用いて評価値の計算を行う．

$$\text{BLEU} = BP_{\text{BLEU}} \cdot \exp(\frac{1}{N}\sum_{n=1} N \log p_n)$$

BP_{BLEU} は翻訳文が参照訳より短い場合に与えるペナルティ (brevity penalty) で，翻訳文の単語数を c，正解文の単語数を r とすると，以下の式で計算される．

$$BP_{\text{BLEU}} = \begin{cases} 1 & (c > r) \\ \mathrm{e}^{1-\frac{r}{c}} & (c \leq r) \end{cases}$$

翻訳文の文字数が正解文よりも長い場合には，$BP_{\text{BLEU}} = 1$ であり，BLEU 値に対して影響を及ぼさない．指数関数の p_n は以下の定義式で記述される．

$$p_n = \frac{\sum_{\text{出力文}} \sum_{\text{正解文}} \text{正解文と一致する N-gram 数}}{\sum_{\text{出力文}} \sum_{\text{正解文}} \text{全 N-gram 数}}$$

p_n は正解文と一致する N-gram 数を数える際，正解文の要素を重複して数えることを回避するための処理を行う．一般的に，$N = 4$ を選択することが多い．本実験においても BLEU は 4-gram を適用する．一般的に BLEU が高い程精度が良いシステムである．しかし，本実験では対話システムの評価であるので，一つの発話に対して応答が一意に定まらないケースは多数存在する．従って，本実験では既存手法と提案手法の BLEU を参考値として確認するために算出する．

4.2.3 WER

Word Error Rate (WER) はスピーチ認識や機械翻訳の分野で用いられるシステム出力の精度を測る手法の一つである．WER は出力文と正解文の類似度を測る際に，出力文と

正解文は異なる文長である可能性があることを考慮している．レーベンシュタイン距離の考えに基づき，以下の定義式により単語レベルでの計算を行う．

$$\mathrm{WER} = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

ここでの，上式の各要素は，出力文の単語を何回操作することで正解文と等しくなるかをそれぞれ表したものである．具体的には，S は正解文からの置換数であり，D は正解文からの削除数，I は正解文からの挿入数，C は正解文の一致数，$N = S + D + C$ は正解文内の単語数である．例えば，出力文「the I had pen」と参照文「I have a pen」が与えられた時，「the」を削除し，「had → have」と置換し，「a」を挿入することで出力文を参照文へと変更できるため，編集操作数が 3 である．これを参照文の長さ 4 で割れば，0.75 という WER 値が求まる．WER は BLEU が開発される前から，音声認識などの評価で広く使用されていた．しかし，WER では参照文と出力文の語順の違いに非常に厳しい評価尺度となっており，例えば「black cat」と「cat black」のような比較的人間に理解しやすい小さな語順の誤りを完全に誤った訳と判定してしまう．このため，WER は BLEU に比べて厳しい評価尺度となる．本実験では，BLEU と同様に WER についても参考値として算出する．

4.2.4 感情語彙の出現頻度

本指標では，提案手法が感情ラベルに従った感情語彙の制御ができているかを検証する．つまり，出力単語の中で感情語彙とみなした単語をカウントし，感情ラベルとの整合性が取れているかどうかを調査する．

$$\mathrm{EmotionWord} = \frac{1}{T} \sum_{t=0}^{T} \mathrm{correct}_t$$

$\mathrm{correct}_t \in \{0, 1\}$ は t 番目の出力文に対し，指定した感情ラベルと同一感情語彙が出力されているかどうかを表す数値である．正解感情ラベルと同じ感情値を持つ感情語彙が出力されている場合 1，出力されていない場合は 0 とする．

5　実験設定

5.1　実験データ

5.1.1　データクローリング

本実験で必要なデータとして，対話システムの Pre-training に必要な大規模対話対データ，ドメインが限定されたトピック対話データ，感情語彙を選定するための感情語辞書が挙げられる．以下に各データの収集方法について述べる．

まず，Pre-training として用いる一般的な対話対として, Twitter API[1] を用いてクロールした対話データ 120 万文対と NTCIR-12[2](NII Testbeds and Community for Information access Research) が推進している STC[3](Short Text Conversation) Japanese Task で使用されたツイートコーパス 42 万文対を収集する．どちらのデータも Twitter 上で行われた対話であり，言語は日本語のものに限定している．

次に，特定のドメインに偏った対話対を収集する．本報告では，Facebook[4] に存在する公開グループから対話対をクロールする．公開グループとは， Facebook 上に存在する同じ趣味や信条を持つ人間同士が集まり交流をする場である．本実験では Facebook Graph API[5] を用いて，日本プロ野球グループとポケモン GO グループの二グループから対話対をクロールする．グループ数を二つに制限した理由として，ニューラルネットによる対話文の学習は一定数の対話データが必要であり，同一トピックの対話文を多数収集することの困難性とドメイン推定実験の簡単化の二点が挙げられる．本実験において上記の二グループを選択した理由として，スポーツやゲーム等の会話は日常の会話とは異なり，感情が発露しやすいトピックであり，本報告の目的として掲げたドメインと感情の両方が出現しやすいグループであると考えられるためである．クロール対象期間はグループの公開日から，2017 年 11 月末迄に会話された対話を対として保存する．クロール対象とする対話として，図 5.2 の用にポストに付随しているコメント（post）にさらに付随しているコメント (reply) をペア (post-reply) として収集する．結果として，野球をドメインとした対話 28, 767 文対，ポケモン GO をドメインとした対話 27, 699 文対をクロールした．

また，ドメイン対話のデータ量をさらに増加させるために，Yahoo ニュース [6] のコメント欄で行われている対話もクロールする．データの収集元ニュースとして，西日本ス

[1] https://developer.twitter.com/en/docs
[2] http://research.nii.ac.jp/ntcir/index-ja.html
[3] http://ntcir12.noahlab.com.hk/japanese/stc-jpn.htm
[4] https://ja-jp.facebook.com/
[5] https://developers.facebook.com/docs/graph-api?locale=ja_JP
[6] https://news.yahoo.co.jp/

図 5.1: Twitter 上での対話履歴

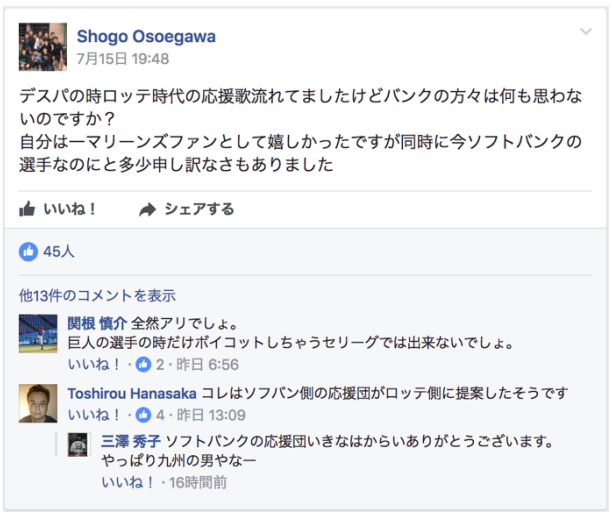

図 5.2: Facebook 上での対話履歴

ポーツ [7] とベースボールキング [8] の RSS からクローラを用いて対話対をクロールする．クロール対象とする対話は Facebook のクロール時と同様にコメントに付属しているコメントを対話文対とする．結果，野球をドメインとした対話 3,708 文対を取得した．

次に External Memory の機構で感情制御のために用いる感情語彙の選定について述べる．2017 年現在日本語の極性辞書として用いられている小林ら [57] の極性辞書と高村ら [58] の極性辞書を用いる．小林らの極性辞書は，用言を中心に収集した評価表現約 5 千件のリストを一部改編し，人手で評価極性情報を付与したデータであり，含まれている単語はポジティブかネガティブかの極性を含んだ単語のみが収録されている．高村らの極性辞書は，岩波辞書から抽出した単語を $-1.0 \sim 1.0$ の値の範囲で極性判定を行っている．本報告では，小林らの辞書の単語群と，高村の辞書の単語群の内 $-1.0 \sim -0.9$ と $0.9 \sim 1.0$ の範囲内に含まれている単語群を合わせた単語群を感情語彙として用いる．しかし，上記の極性辞書は通常の辞書に掲載されているような一般的な語句についてしか載っておらず，実際のツイートデータで用いられているような現代的な感情語彙についてはカバーできていない (例：“ウザい”，“キモい”，等) ．よって，そのような感情語彙をカバーするために，ツイートから新たに感情語彙を取得し，それを既存の感情語彙に加えて使用することとする．具体的には，まず事前に感情が含まれているであろうツイートをクロールする．クロール対象として，怒っていることを表す絵文字のバイト文字 `\xf0\x9f\x92\xa2` （ネガティブな感情を有するツイートとみなす）と，好意を表す絵文字のバイト文字 `\xf0\x9f\x92\x95` （ポジティブな感情を有するツイートとみなす）が含まれているツイートを検索・収集する．結果として，二つの絵文字がそれぞれ含まれて

[7] https://headlines.yahoo.co.jp/rss/nishispo-c_spo.xml
[8] https://headlines.yahoo.co.jp/rss/baseballk-c_spo.xml

いるツイートを約 20 万文ずつ抽出した．この二つのツイートデータに対し，word2vec でそれぞれ単語分散表現を獲得する．word2vec の学習は CBOW モデルを使用し，分散ベクトルの次元は 200 ，ネガティブサンプリングを用いる．作成された単語分散表現に対し，人手で収集した現代感情語彙 57 単語をシードとする．このシード毎に単語分散ベクトルをモデルから計算し，このベクトルと類似する単語上位 15 件を新たに感情語彙として記録する．この際，既存の感情語彙との重複はないものとする．結果として，ネガティブな単語 1,621 件，ポジティブな単語 2,666 件を抽出した．本報告ではこれを感情語彙として使用する．

5.1.2 データの前処理

5.1.1 節でクローラによって収集されたテキストデータからノイズを除去したものを実験データとする．本節では，具体的なノイズ除去について記述する．

本実験で用いる対話データの収集元が Twitter や Facebook 等のソーシャルメディアであるので，投稿文に含まれている URL，引用ツイート記号，ハッシュタグは除去する．また，日本語の形態素解析をする上で，ノイズとなるような改行，顔文字，絵文字，重複記号（“www” や “！！！！！ ” など）を除去する．重複記号については，4 回以上続いている文字を一文字に集約する．

日本語の形態素解析器として，MeCab[9] ，日本語辞書として mecab-ipadic-NEologd[10] [59, 60] を用いる．mecab-ipadic-NEologd は，Web 上の多数の言語資源から得た新語を追加することでカスタマイズした MeCab 用のシステム辞書である．形態素解析された対話文の内，一文当たりに含まれる単語数が 30 を超えるものは，seq2seq の学習時間が増大するため実験データから除去した．

以上の前処理の結果，Pre-training 用対話文対 1,325,667 件，Fine Tuning 用特定会話ドメイン対話文対 58,583 件が得られた．本報告では，これらのデータを実験データとして使用する．

5.2 パラメータ設定

5.2.1 提案手法の学習設定

本提案システムの具体的な実験設定について述べる．本システムの学習は，大量の一般対話対データで Pre-training を行い（図 3.8），その重みを使用して特定ドメイン対話対データで Fine Tuning （図 3.9）を行う．使用する単語辞書は予め使用するデータを一つに集約したものから出現頻度に基づいた選定を行う．具体的には，コーパスの作成時に語

[9] http://taku910.github.io/mecab/

[10] https://github.com/neologd/mecab-ipadic-neologd

彙数の上限値を 45,000 単語として 3.5 節で示した word2vec を用いた単語の置き換えを行う．5.1.1 節で抽出した感情語彙の内，実際にコーパス 45,000 単語内に含まれている感情語彙の数は 1,387 単語であった．

提案手法の学習方法として初めに，seq2seq 及び External Memory と tweet2vec に分けて Pre-training を行う．seq2seq 及び External Memory は学習データとして，5.1 節で収集した一般ドメイン対話コーパスを用いる．開発データとして，ランダムに 1,000 件分をコーパスから抽出し，残りのデータを学習に利用する．学習エポックは 100 epoch，バッチサイズ 200 ，語彙数 45,000，単語次元数 256 ，LSTM の隠れ層次元数 512 ，感情ラベル数 3 (positive, negative, neutral) ，感情ラベル次元数 64 とする．最適化関数は Adam [61] を用い，学習率は 0.001，Dropout Rate [62] は 0.2 とする．また，正則化手法として Weight Decay を使用する．学習終了後に各エポック毎のロスを計算し，バリデーションセットのロスが低いモデルを Pre-training 時の初期値に設定する．

tweet2vec は 5.1 節で収集した特定会話ドメイン対話コーパスを用いて学習を行う．テストデータとして，各トピック 100 件，計 200 件をランダムで抽出し，残りをトレーニングデータとして使用する． 学習エポックは 30 epoch，バッチサイズ 64 ，入力文の最大サイズは 145 ，文字の Look-up Table の次元数 150 ，C2W の次元数 500，単語次元数 500，推定トピック数 2，学習率は 0.01， Gradient Clipping は 5 に設定する．

Pre-training の終了後，システムの Fine Tuning を行う．学習エポックは 100 epoch，バッチサイズ 100 ，語彙数 45,000，単語次元数 256 ，LSTM の隠れ層次元数 512 ，ドメインラベル数 2 (野球，ポケモン GO) ，ドメインラベル次元数 64 , 感情ラベル数 3 (positive, negative, neutral) ，感情ラベル次元数 64，最適化関数は Adam ，学習率は 0.001，Dropout Rate は 0.2 とする．単語辞書は Pre-training で使用した辞書を使用する．感情語彙は Pre-training 時と同じ 1387 単語とする．Fine Tuning のモデルの決定は，Pre-training の決定方法と同様，バリデーションロスが低いエポックのモデルを適用する．適用されたモデルを，本実験での提案モデルとする．

5.2.2 比較手法の学習設定

提案モデルと比較するための既存モデルとして，提案モデルから会話ドメイン推定部と感情制御部を除いた，通常の Bi-Directional Sequence-to-Sequence を既存手法として設定する．使用データは提案モデルがトレーニングデータとして使用した全てのデータをトレーニングデータとして使用する．既存手法のモデルの単語次元や隠れ層次元数，最適化関数，学習率，Dropout Rate は提案手法と同様である．また，Vinyals [36] らの実験において Attention 機構は seq2seq による対話の精度に寄与しなかったことから，本実験では seq2seq に対し，Attention 機構の実装を行わない．

5.2.3 テスト時の設定

テスト時は，提案手法と既存手法共にビーム幅 5 のビームサーチによって出力を生成する．両手法の語彙数は低頻度語を unknown tag (<unk>) で置き換えている．出力文に <unk> が出現することによって，文意の著しく損ねてしまう．従って，<unk> ラベルが出現した場合は，次に生起確率の高い単語を出力単語として選択する．

5.2.4 人手評価の設定

提案手法と既存手法の出力の人手評価は人手で用意したテストセット 600 文を用いて評価する．人手評価に使用したデータは Task1, Task2 共に 300 件ずつ（内 150 件は野球ドメイン発話，残り 150 件はポケモン GO ドメイン対話）計 600 件を人手で作成する．このデータは各ドメインに沿った発話となっており，一般的な会話ドメインの発話（「こんにちは」や「おやすみなさい」等）を除去している．Task1 では，入力に対して提案手法の positive もしくは negative な応答どちらかと既存手法の応答とを提示し，その出力それぞれを評価する．各システムの出力はアノテートの公平性のためランダムシャッフルされており，どちらが提案手法か既存手法であるかは評価者からはわからない前提で評価を行う．Task2 では，入力に対して提案手法の positive，negative，neutral のいずれかの出力を提示し，その出力に対して実際の感情値を評価する．実際にモデルに入力した感情ラベルは評価者には伏せた状態で評価を行う．手法に対する事前知識のない本研究室の学生 5 名に評価してもらい，4 章の各評価値の平均を本実験の評価値とする．

6　実験結果

6.1　ロス関数の推移

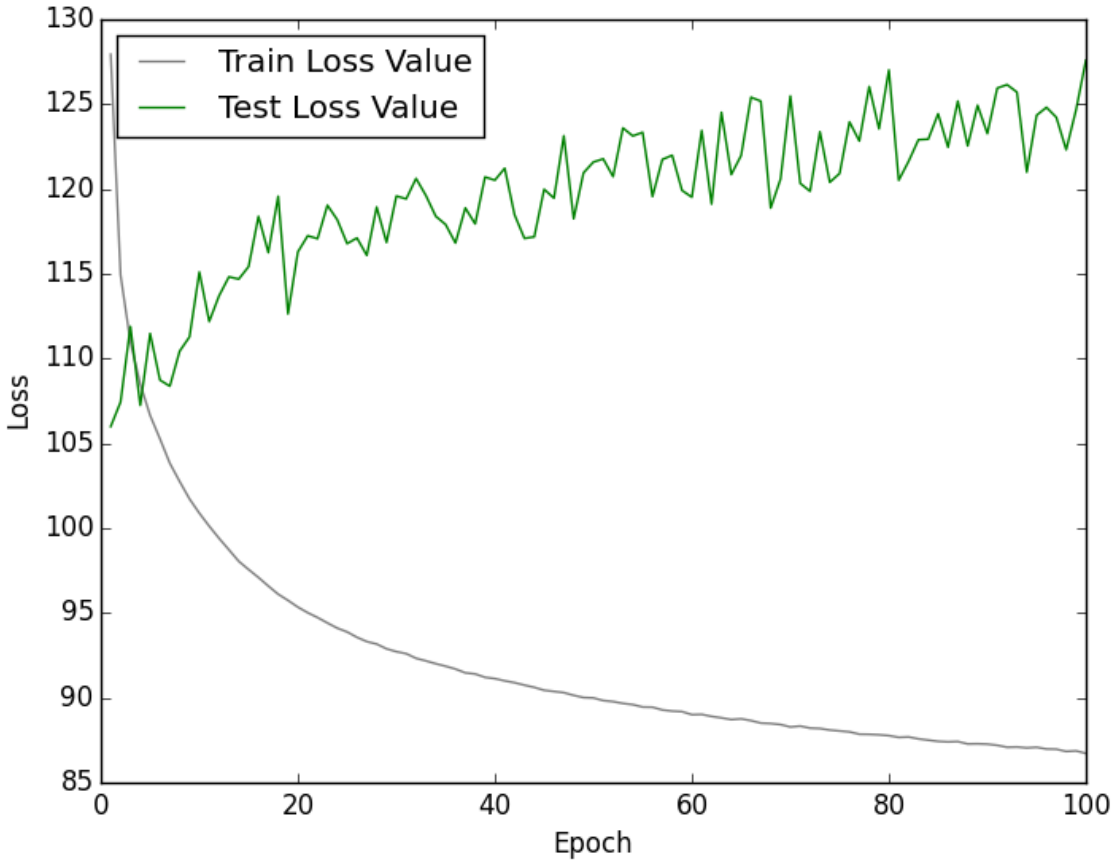

図 6.1: Pre-training 時のエポック毎のロスの値

一般ドメイン対話コーパスを用いて， Pre-training した結果を示す．図 6.1 に Pre-training 時のエポック毎のモデルのロスをプロットしたグラフを示す．図から，トレーニング時のロスは一貫して下がり続けているが，テストデータのロスは全体的に上昇していることが見て取れる．これは，機械翻訳等のタスクとは異なり，対話データは一つの文に対して複数の回答が考えられるために，テストデータに対して学習が進んでもロスが下がるという保証がないことに起因するためであると考えられる．また，20 epoch 以降のテストロスが上昇していることから，本実験では Fine Tuning として用いるモデルの学習前の初期値を，Pre-training の 20 epoch の重みに設定する．

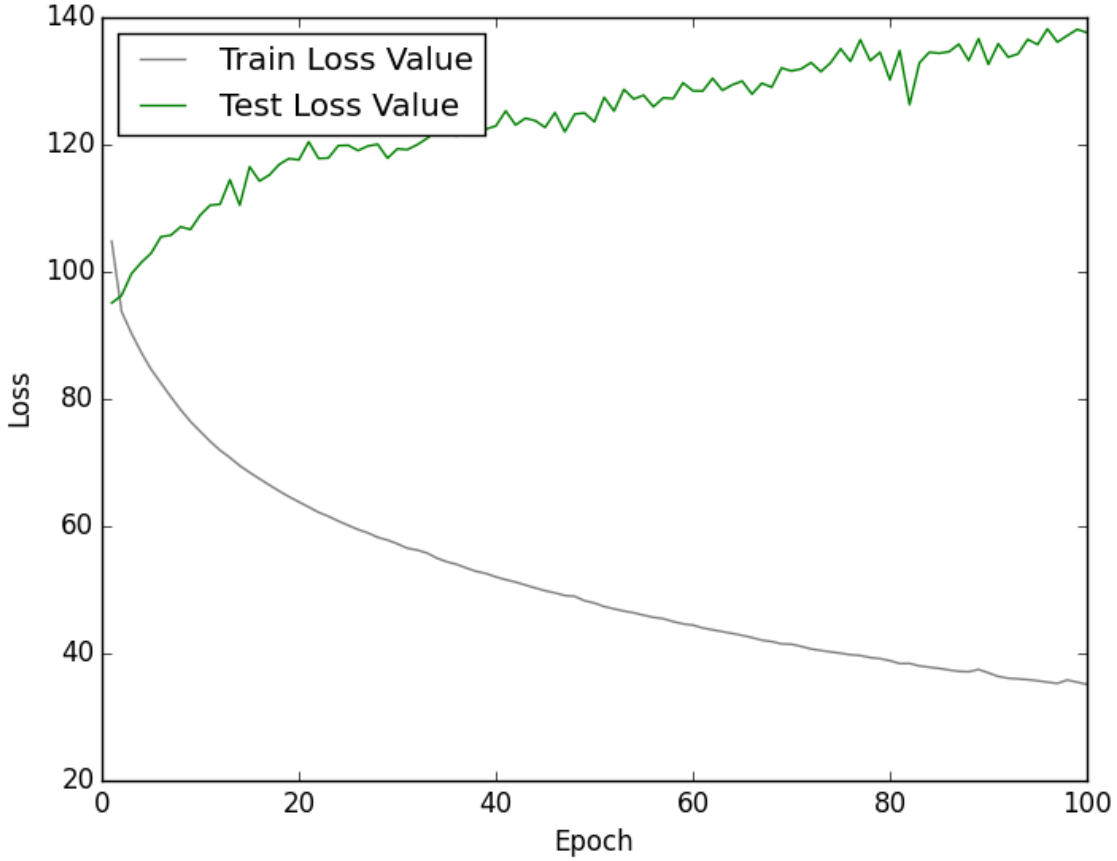

図 6.2: Fine Tuning 時のエポック毎のロスの値

特定ドメイン対話コーパスを用いて， Fine Tuning した結果を示す．図 6.2 は Fine Tuning 時のエポック毎のロスを表している．6.1 節と同様に， Fine Tuning 時もデータのロスの増減は同様の傾向を示している．原因についても，同一の理由であると考えられる．本研究では，提案手法の重みを 20 epoch のものを適用する．

6.2 自動評価の結果

表 6.1: Tweet2vec の分類精度

評価値	分類精度
Accuracy (Precision@1)	0.89

本節では，既存手法と提案手法の自動評価による結果を比較する．本提案手法では tweet2vec を会話ドメインの推定器として使用しており，その推定精度について tweet2vec 単体での精度を測定する．tweet2vec 単体の精度評価はテストセットを用いて分類器の二

値分類の精度を測定する．具体的には，学習で使用しなかった 200 文を用いて野球とポケモン GO のトピックのどちらであるかの二値分類精度を測定する．tweet2vec による会話ドメイン特定精度は表 6.1 の通りであった．精度値の結果から，全会話ドメイン数が 2 の場合，約 90 % の割合でドメインを特定することに成功した．

表 6.2: 自動評価指標の結果

指標	既存手法	提案手法
BLEU	1.39	1.54
WER	197.28	197.36

表 6.3: 感情語彙の出現頻度の自動評価結果

指標	提案手法の精度
EmotionWord	0.946

表 6.2 に，応答文の自動評価指標の結果について示す．BLEU と WER は学習の 20 epoch でのテストセットによる評価値を表に示している．表より，BLEU と WER は既存手法・提案手法に大きな差異は現れなかった．これは，6.1 節で述べた通り，発話文に対する応答文の評価は文脈に依存しており，且つ複数の回答が存在しうるため，自動評価手法では正確な精度が測ることが困難である [37, 46]．各指標のエポック毎のバリデーションセットによる評価値については図 7.2 ～ 7.5 に示す．

また，表 6.3 に感情語彙の出現頻度を表す EmotionWord の評価値を示す．データは Task2 で使用したデータの内，感情ラベルとして “positive” ラベルと “negative” ラベルを入力したデータセット 204 件を対象としている．データセット 204 件の内，指定した感情ラベルと同一の感情語彙が出力されている文は 193 件存在した．すなわち，EmotionWord の値は 0.946 であった．

6.3 人手評価の結果

表 6.4: Task1: 既存手法と提案手法の人手評価結果

指標	既存手法 (Sequence-to-Sequence)	提案手法
Fluency	**0.995** (±0.006)	0.955 (±0.023)
Consistency	**0.773** (±0.094)	0.753 (±0.127)
DomainConsistency	0.109 (±0.044)	**0.890** (±0.044)
Sentiment	0.282 (±0.133)	**0.717** (±0.133)

次に，人手評価による既存手法と提案手法の精度比較を行う．表 6.4 に Task1 の 4 つの指標の評価値を示す．各指標の信頼区間は 95% に設定している．表より，流暢性と一貫性は，やや既存手法が提案手法に比べ値が高い結果であったが，会話ドメイン一貫性と感情の豊かさでは，提案手法が既存手法に比べて評価値が大きく高いという結果となった．このことから，提案手法では既存手法の流暢性や一貫性を大きく落とすことなく，より会話ドメインと感情を加味した応答が生成されていることがわかる．

表 6.5: Task2: 感情制御に対する人手評価結果

指標	提案手法の精度
EmotionTag	0.645 ($\pm$0.023)

表 6.5 に Task2 の指標である EmotionTag の評価値を示す．指標の信頼区間は 95% に設定している．評価値から，本提案モデルは感情語彙を制御することで約 65 % 程度の出力感情を制御可能であるという結果が得られた．

6.4 考察

tweet2vec による会話ドメインの推定精度は，表 6.1 に示す通り約 90 % の割合でドメインを特定することに成功した．これは，テキストの二値分類の結果であるので，あくまで参考値であるが客観的には高い精度であると考えられる．データの都合上二値分類のみの結果であるが，tweet2vec の提案論文 [31] では約 200 万のツイートを 2,039 種類のハッシュラベルにクラス分類している．従って，tweet2vec 自体はより多数のドメインへの対応が可能であると考えられる．今後，特定ドメイン対話データが増加してドメイン数が増加した場合，分類精度を再度測定する必要がある．

表 6.6: Fleiss' Kappa

指標		Fleiss' kappa
Fluency	既存手法	0.2115
	提案手法	0.4464
Consistency	既存手法	0.4051
	提案手法	0.3685
DomainConsistency		0.7782
Sentiment		0.4587
EmotionTag		0.5198

また，アノテートの信頼性を確認するため，アノテートされたデータを評価者間の合意の

表 6.7: 提案手法と既存手法の出力の比較（会話ドメイン：野球）

入力	Model	出力
レギュラー取って間もないですよね	既存手法	そうなんですか?
	提案手法	ただ、巨人では『下克上』だと思いますけど、その価値が良いのではありません!

信頼性を評価するための統計的尺度である Fleiss' kappa [63] を算出した．各指標の Fleiss' kappa 値は表 6.6 に示す．表 6.6 から，DomainConsistency と EmotionTag は一致率が高く，Fluency と Consistency は一致率が低い結果となった．これは，DomainConsistency と EmotionTag は比較的評価が簡単なタスクであり，Fluency と Consistency は文法や前の文脈を観察する必要があるので難しいタスクであったと言える．DomainConsistency と同様に AB テスト方式である Sentiment の Fleiss' kappa 値が低い理由として，既存手法の出力にも提案手法と同様の感情が含まれている文があるので，より感情の発露が見られる文を選択することが困難であるケースが存在するためであると考えられる．

Task1 の人手評価 (表 6.4) の結果，本提案手法の Fluency と Consistency は既存手法の出力と同程度の精度であることがわかった．これは，本提案手法が既存手法と比べて出力の精度を落とすことなく応答生成ができたと考えられる．本提案手法の Fluency と Consistency が既存手法と比べて，少しだけ低い値を示した理由として，既存手法では汎用的な応答を生成することが原因として考えられる．表 6.7 に同一の入力を提案手法と既存手法に与えた時の出力例を示す．表から，提案手法ではドメインを考慮した出力を生成しているが，入力文に対する適切な応答とはなっておらず文意が理解できない．従って，このケースだと提案手法の Fluency と Consistency は “0” と判定される．一方，既存手法の出力はドメインを考慮していない一般的な応答が生成されているが，入力に対しての応答として文意は通っているので Fluency と Consistency は有りと判定される．表 6.7 のようなケースが複数存在するために，提案手法の Fluency と Consistency が既存手法と比べて低い値を示したと考えられる．また，本提案手法の DomainConsistency と Sentiment が既存手法に比べて大きく精度が上回っていた原因として，表 6.7 のように提案手法の出力にはドメイン依存の単語 (“巨人”) や感情語彙 (“良い”) が既存手法の出力に比べて含まれていることが多いからだと考えられる．出力単語に上記の単語が含まれていることは評価者にとって判定がしやすくなる要因であることを示している．このことは，表 6.6 が示すように Fluency と Consistency に比べて DomainConsistency と Sentiment の Fleiss' kappa 値が高いという要因の一つであると考えられる．

表 6.8 に既存手法と提案手法それぞれの出力の一例を示す．表の提案手法の出力単語の内，赤色の単語はポジティブワードであり，青色はネガティブワードである．既存手法では，入力文に対して文意の通った応答を返しているが，会話ドメインに沿った応答ではなく一般的でありきたりな応答を返していることがわかる．しかし，提案手法では入力の

表 6.8: 提案手法と既存手法の出力例 1

入力	Model		出力
菅野は日本のエースになるよ	既存手法		そうなの?
	提案手法	Positive	菅野は良いですよね！
		Neutral	菅野は故障が多いので、来年も、頑張って欲しいですね
		Negative	菅野は無理ですね！
金本監督は若手をどんどん育てて欲しいですね。	既存手法		私もそう思うけど、私は好きです。
	提案手法	Positive	巨人の監督に期待してます ね
		Neutral	巨人の監督になったら、そうですね 。
		Negative	巨人の外国人選手も、怖いですね 。
今年は横浜が帰って来ますよ。	既存手法		今日はありがとうございます!
	提案手法	Positive	横浜はまだ若いですね 。
		Neutral	横浜はまだゲーム差があるけどね。
		Negative	横浜はまだ無理ですね。
平田選手は怪我に気をつけて頑張って欲しいですね！	既存手法		ありがとうござい ます！
	提案手法	Positive	そうですね。期待してますが、ドラゴンズの投手ではないかどうかですか。
		Neutral	そうですね。もうその時は楽天に勝ち越していますから 。
		Negative	そうですね。それが難しいみたいですよ 。
大谷選手のメジャー移籍の可能性はあります	既存手法		俺は今からの方が好きです
	提案手法	Positive	そうなんですか!それは、期待してるんですよね!
		Neutral	そうなんですか!それは知りませんでした!
		Negative	そうなんですか!それは、どこの球団にも失礼ですね!
中田は先発にいらないよね	既存手法		それはそれでいいですよね
	提案手法	Positive	藤浪は来季良いですよ！
		Neutral	藤浪は糸井かと思っていたけど・・・
		Negative	藤浪は普通に打つ方が問題にあるけど・・・
この時ばかりは二キロタマゴと交換してあげたかったですわら	既存手法		あ、よかったです！
	提案手法	Positive	金コイは個体値良かったですね
		Neutral	金コイとイーブイでしたね
		Negative	金コイはキツイですね
ミニリュウ、うらやまし過ぎます！	既存手法		明日は明日にしてくれてもいいですか?
	提案手法	Positive	時間はランダムですが、必ずミニリュウが出るようなことがあってタイミングが合えば必ず get 出来るんです
		Neutral	時間はランダムですが、必ずミニリュウが出るような感じでした
		Negative	時間はランダムですが、ミニリュウが弱い時にはミニリュウが出てきてくれました

ドメインに沿っており，且つそれぞれの感情ラベルに従った感情語彙を元に出力を生成している．以上から，表 6.8 の出力においては，本報告の目的である会話ドメインと感情制御を考慮した応答生成が達成されていることが分かる．また，表 6.8 の上の一例での “neutral” ラベルの出力において，ポジティブワードとネガティブワードが両方出てきている例が存在する．これは，学習の段階で，External Memory を学習する段階で，正解応答文が “positive”，“negative”，“neutral” のどの感情に属するかを単純に感情語彙の個数で振り分けており，ポジティブワードとネガティブワードの語彙数が同数の場合，自動的に “neutral” にラベル付けされることに起因されると考えられる．“neutral” にポジティブとネガティブの感情語彙が同数ある文が多数含まれることにより，同数であるという生起条件が “neutral” のラベルを入力した際に学習されていると考察される．

表 6.9: 提案手法による出力例 2（会話ドメイン：野球）

入力	Model		出力
今年は勝てるように頑張って応援します！	提案手法	Positive	今年は、負けたことがないので、ちょっと応援してます
		Neutral	今年は、負けても負けても、頑張って欲しいです
		Negative	今年は、負けてもないので、ちょっと怖いです。

表 6.10: 提案手法による出力例 3（会話ドメイン：野球）

入力	Model		出力
カープの場合、ロペスはどうでしょうか？ 移籍したこと抜きで。	提案手法	Positive	監督もチャンスであるのではないですか？
		Neutral	監督は打撃が、日本には、巨人が入りして、そんなことがあると思います。
		Negative	監督は厳しいですが、怪我の離脱がしているみたいです。

さらに，表 6.9, 6.10 に別のテスト入力発話に対する提案手法の出力を示す．表 6.9, 6.10 の入力発話の会話ドメインはどちらも野球のドメインである．表 6.9 では，野球のドメインではあるが発話内に特に野球でしか用いられないような用語が使われていない発話が入力になっている．そのような入力に対する出力は全体的に文法の崩れが少ないことが表 6.9 からわかる．これは，一般ドメインでも含まれるようなコーパス内に多く観測される単語で構成された文であるので，学習しやすいことが考えられる．一方，表 6.10 では “ロペス” や “カープ” など非常に野球のドメインに依存した発話であり，その入力に対する出力は助詞などの細かな文法が崩れている．文法が崩れる原因として，上記のような単語はコーパス内では低頻度語であり，学習データに含まれる頻度が小さいので単語の正確な生起確率が学習されづらいことが挙げられる．

また，別の要因として、External Memory による感情語彙の出力への挿入がシーケンスの出力状態に悪影響を及ぼしていることが考えられる．これは，Emotional Chatting

表 6.11: 学習に使用した対話データの感情ラベル毎の応答文数

対話データ (収集元)		テキスト数 (pair)	positive	neutral	negative
一般ドメインデータ (Twitter)		1,081,500	218,322	760,451	102,727
特定ドメインデータ (Facebook)	野球	24,666	6,011	16,623	2,032
	ポケモン GO	22,734	7,633	13,299	1,802

Machine [46] の論文でも議論に上がっている内容である．具体的には，External Memory はデコード部に直接影響を与えるモデルであり，直接的に感情制御をコントロールすることが可能であるが，LSTM の出力に対して直接操作を加えるため，通常の出力層に比べて文法が崩れやすいことが報告されている．以上の二つの理由から，提案手法の出力は既存手法に比べて文法を崩しやすいと考えられる．

また，Task2 の提案手法の出力 300 文 の中で，“positive” ラベルと “negative” ラベルに対する出力からポジティブワードとネガティブワードをそれぞれ抽出し，実際に出力されている感情語彙を観測した．表 6.12, 6.13 にそれぞれの感情ラベルに対して生起した感情語彙を示す．表 6.12 に出力されているポジティブワードの中には，野球ドメインで用いられる “ファン” や “応援” ，“生え抜き” のような単語や，ポケモン GO ドメインで用いられる “レア” や “進化” ，“強化” のような単語が含まれていることがわかる．このことから，“positive” ラベルで共起される出力にはドメインに沿った感情語彙が生起していると考えられる．表 6.13 に出力されているネガティブワードも同様に，“悪い” や “難しい” などのネガティブな感情を想起させる単語が生起していることが確認される．しかし，ネガティブワードの中には “来る” や “入る” などの通常の単語であったり，“ビックリ” ，“羨ましい” のようなポジティブに近い単語もネガティブワードとして生起している．このように，ネガティブワードの生起が崩れている原因として，2 つの原因が考えられる．

一つは、そもそもネガティブ文が Facebook データセットにポジティブ文ほど含まれておらず，ポジティブワードに比べて十分な学習に至っていないことが考えられる．表 6.11 に実験で使用した対話データの応答文について感情ラベル別に文数をカウントした結果を示す．表から，全体的にポジティブ文に比べてネガティブ文の割合が少なく（Twitter データは全体の約 9.5 %，Facebook データは全体の約 8.1 %），特に Facebook からクロールしたポケモン GO 対話データはネガティブ文はポケモン GO 対話データ全体の内，約 7.9 % であった．これはクロールした SNS ドメイン特有の問題である．具体的には，Facebook は話者の本名が明らかな中で会話を行うので，他人を誹謗中傷するようなネガティブな文を投稿することは印象が悪く，敬遠されることが多い．従って，Facebook の対話データは同意や応援などの比較的ポジティブな投稿が多い．コーパス内にネガティブな応答が少ないことによって，システムのネガティブな応答の学習が不十分になったと考

えられる．

もう一つの理由として，感情語彙の選定の際にネガティブワードを上手く選択できていないケースが考えられる．上で述べた “ビックリ”，“羨ましい” といった単語は，クロールしたネガティブツイート（怒っていることを表す絵文字のバイト文字 `\xf0\x9f\x92\xa2` が含まれているツイート群を指す）を word2vec によって単語分散表現化し，感情語彙との単語間類似度の高い単語を選択した際に選ばれた単語である．先の章でも述べた通り，この操作は既存の感情語彙には含まれていない現代的な感情語彙を取り出す意図で行なった操作である．実際に，この操作は現代的な感情語彙の取り出しには成功しているが，同時にノイズとなるような単語も感情語彙として収集される．これは，バイト文字 `\xf0\x9f\x92\xa2` が含まれているツイート群が単純に怒っている内容を表しているツイートだけではないことが一因となっている．例えば，「ありがとう（怒りの絵文字）」といったような，通常のポジティブな単語に怒りの絵文字を付与することで逆説的な意味やユーモアを相手に印象づけることができる．これは Twitter のような他者との会話を行う SNS 上でよく見られる表現である．上記のような日本語のネット特有の表現が存在することから，ツイートデータにポジティブな単語も含まれていることが word2vec の分散表現の学習に影響を及ぼしたと考えられる．データのクロール先と感情語彙の選定方法が大きくシステムの感情制御に依存することから，両者の操作については考慮の余地があると考えられる．

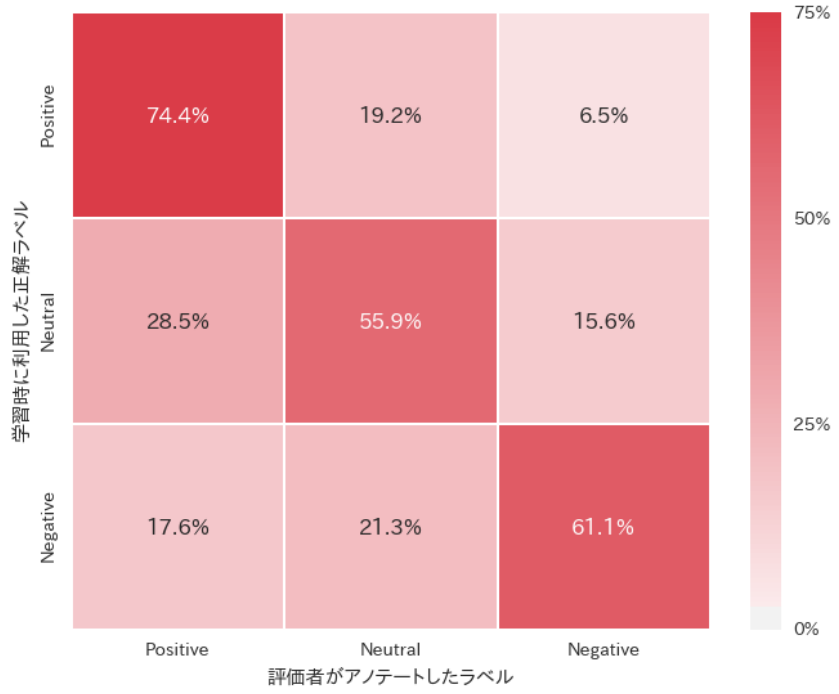

図 6.3: Task2: 学習に使用した感情ラベルと評価者がアノテートした感情ラベルの整合性

表 6.12: Task2 の提案手法の “positive” な出力に含まれている感情語彙一覧（一部抜粋）

出現した感情語彙	出現頻度（回数）
ありがとう	16
いい	11
凄い	8
楽しみ	7
良い	7
期待	7
ファン	5
嬉しい	5
強化	4
言っ	4
良かっ	4
おめでとう	3
好き	3
一番	3
可愛い	3
レア	2
復活	2
大好き	2
応援	2
立て直す	1
最高	1
貴重	1
生え抜き	1
よかっ	1
素晴らしい	1
進化	1
地道	1
幸い	1
嬉しかっ	1
若い	1
面白い	1
楽しい	1

表 6.13: Task2 の提案手法の “negative” な出力に含まれている感情語彙一覧（一部抜粋）

出現した感情語彙	出現頻度（回数）
無理	11
失礼	9
難しい	8
田舎	6
大変	5
ビックリ	5
羨ましい	4
打つ	3
悪い	3
すごい	3
寂しい	2
来る	2
入る	2
できる	2
恥ずかしい	2
マジ	2
怪しい	2
暑い	1
厳しい	1
がっかり	1
余計	1
駄目	1
扱い	1
しない	1
弱い	1
最低	1
キツイ	1
やめ	1
怖い	1
勘違い	1
悲しい	1
痛い	1

表 6.14: Negative 出力を Positive として見なされた例

誤答者数 (人)	Negative 出力
5	羨ましいです!私も地元の本拠地は球場でポケ活してます!
5	僕はすごいですよね
5	はい!アメの方がすごいです!
5	野生のラプラスは 2 匹目をゲット出来ました。。。私も初めてビックリしました。
4	私も初めてでしたビックリしました!
4	私も、頑張ります
4	僕も、まだ 1 体もすごいです!
3	そうですよね (笑) でも、メジャーに入るのは時間の問題ですからね (笑)
3	オールスター出場にふさわしい活躍を見せてくれると思いますが、ちょっと空気読めますね!
3	もう野生では、ポリゴンは羨ましいです
3	ホントですよね。やった時はビックリでした!
3	はい。ありがとうございます。そうですよね。
3	そーですよね!ジムも少ないんですけど、ちょっと羨ましい!
3	そう!マジですか?
3	私もビックリしました!
3	ポケモンを捕まえることができるから、大丈夫かも

感情制御に対してより深く考察するため，Task2 のデータで学習に使用した感情ラベルと評価者がアノテートした感情ラベルの関係性を図 6.3 で示す．各セルの値は，5 人の評価者がつけたラベルと実際の学習で使用したラベルの関係をカウントした数値を各学習ラベル数に対する割合で示している．図からわかる通り，学習時はニュートラルとして学習した文は，評価者からポジティブやネガティブであると学習とは異なる判定値を付けているケースが多数存在する．逆に，ポジティブやネガティブの文として学習した文が，評価者からニュートラルな文として判定値を付けているケースも多数存在する．これは，表 6.11 からもわかる通り，ポジティブやネガティブなテキストがニュートラルのテキストに比べて少ないことが起因している．また，学習時のニュートラル文かどうかの判定は感情語彙の個数で機械的に振り分けられているので，実際はニュートラルではない文も含まれていることも原因の一つであると考えられる．

また，注目すべき点として，学習時はネガティブな出力として学習されている文が評価者から見るとポジティブな文として評価されているケースがテストネガティブ文の内 17.6 % 程度存在していることが挙げられる．上記のケースに当てはまったケースの内，5 人の評価者の内過半数が上記のケースである出力を表 6.14 に示す．表 6.14 のテキストでは，本来ポジティブな単語として使われることの多い “羨ましい” や “できる”，“すご

い” がネガティブワードとして入っている．これは，文脈依存の語彙であり，ポジティブネガティブどちらでも使用されるケースがあるため，感情語彙の選定の際に既存の極性辞書のスコア付けが困難な単語であることが考えられる．また，感情語彙として選定すべき “問題” や “(笑)”，“少ない” が感情語彙として登録されていなかった．これは，文章の内容によってポジティブかネガティブかの極性が変化する感情語彙であるため極性を断定できず，既存の感情語彙として登録されていなかったものと考えられる．従って，本システムの感情制御の精度を向上させるためには，上記のような文脈に依存した感情語彙の選定を考慮する必要がある．

7　結論

本報告では，既存のニューラル対話モデルが汎用的な応答を生成してしまう問題の解決を目的とし，会話のドメインと感情の両方を考慮した対話システムの実装を行なった．対話システムに入力発話と応答において表現してほしい感情ラベルの二つを入力として，入力発話のドメインと感情ラベルを考慮した発話生成を行うシステムを実装した．本提案システムのドメイン推定部は tweet2vec を，会話の生成器は Sequence-to-Sequence Model を，感情ラベルの入力には Speaker Model を，感情語彙の制御には External Memory を用いた．上記のモデルを一つのシステムとして統合し，そのモデルを Twitter と Facebook からクロールしたデータを用いて，重みの学習を行なった．提案手法の学習は，Pre-training と Fine Tuning の二段階の学習を行なった．Pre-training では Twitter から収集した一般ドメイン対話データを用いて応答生成器と感情制御器の学習を行い，同時に Facebook から収集した特定ドメイン対話データを用いてドメイン推定器の学習を行なった．Fine Tuning では Facebook の特定ドメインデータを用いてドメイン推定器以外の重みの再更新を行なった．本実験では，提案手法の出力を既存手法の出力と比較することで，提案手法の有用性の検証と出力の分析を行なった．具体的には，出力の文法が正しいかを評価する流暢性，入力発話との整合性を評価する一貫性，出力が会話のドメインを考慮しているかを評価する会話ドメイン整合性，出力に感情が表現されているかを評価する感情の豊かさの 4 つの指標をメインとして評価した．

結果として，本提案手法は既存手法に比べて流暢性と一貫性のほぼ同等の精度で，会話ドメインと感情を出力に表現することができた．具体的には，既存手法では汎用的な応答生成がなされているが，提案手法ではドメイン依存の単語や感情ラベルに従った感情語彙を出力することに成功した．これは，提案手法が会話ドメインと感情を入力から考慮し，出力に表現することに成功しているためであると考えられる．

一方で，提案手法の出力には文法が崩れてしまうものが複数存在した．これは入力発話にドメイン依存の低頻度語が含まれている場合，低頻度語の学習ができていないことが原因として考えられる．低頻度語の学習が困難であることは，既存手法でも同様の問題として存在するので低頻度語の学習については別途考慮する必要がある．また，生成された感情語彙を抽出したところ，ポジティブな感情語彙に比べてネガティブな感情語彙の方がノイズとなる単語が複数入っていた．原因として，クロールできたネガティブ文のデータ量が少量であることとネガティブである感情語彙の選定時の word2vec にノイズが含まれていることが挙げられる．より精度の高い感情制御を達成するには，感情付き対話データの量を増加させるか，より正確な感情語彙の選定方法を考慮する必要があると考えられる．上記のような点を，システムの改善策として今後の研究に生かす予定である．

謝辞

本報告の全過程を通して多大な御教授，御援助を賜りましたマルチメディア工学専攻ビックデータ工学講座の鬼塚真教授に厚く感謝の意を表します．また，本報告を遂行し，本報告書作成にあたり，日々，懇切丁寧な御指導，御協力を頂きました本研究室の荒瀬由紀准教授，佐々木勇和助教，データビリティフロンティア機構の Chenhui Chu 特任助教に心より御礼を申し上げます．そして，研究テーマ発表会等を通じて御討論，御支援を頂きました鬼塚研究室諸氏に感謝致します．本報告は，Microsoft Research Asia および 株式会社コトバデザイン の助成を受けたものです．

参考文献

[1] I. Sutskever, O. Vinyals, and Q.V. Le, “Sequence to sequence learning with neural networks,” In Proceedings of NIPS, pp.3104–3112, December 2014.

[2] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.Y. Nie, J. Gao, and B. Dolan, “A neural network approach to context-sensitive generation of conversational responses,” In Proceedings of NAACL-HLT, pp.196–205, May 2015.

[3] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, “A diversity-promoting objective function for neural conversation models,” In Proceedings of NAACL-HLT, pp.110–119, June 2016.

[4] T. Hofmann, “Probabilistic latent semantic indexing,” In Proceedings of SGIR, pp. 50–57, August 1999.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” Journal of Machine Learning Research, vol.3, pp.993–1022, May 2003.

[6] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical dirichlet processes,” Journal of the American Statistical Association, vol.101, pp.1566–1581, November 2006.

[7] M. Udell, C. Horn, R. Zadeh, and S. Boyd, “Generalized low rank models,” arXiv preprint arXiv:1410.0342, October 2014.

[8] D. D. Lee, and H. Sebastian. Seung, “Learning the parts of objects by non-negative matrix factorization,” Nature, vol.401, no.6755, pp.788–791, October 1999.

[9] H. Harold, “Analysis of a complex of statistical variables into principal components.,” Journal of educational psychology, vol.24, no.6, pp.417–441, September 1933.

[10] T. Mikolov, W. Yih, and G. Zweig, “Linguistic regularities in continuous space word representations.,” In Proceedings of HLT-NAACL, pp.746–751, June 2013.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint, arXiv:1301.3781, September 2013.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” In Proceedings of NIPS, pp.3111–3119, October 2013.

[13] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," arXiv preprint arXiv:1309.4168, September 2013.

[14] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," In Proceedings of EMNLP, vol.32, pp.1532–1543, October 2014.

[15] T. Mikolov, and Q. V. Le, "Distributed representations of sentences and documents," In Proceedings of ICML, vol.32, pp.1188–1196, June 2014.

[16] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," In Proceedings of ACL, pp.417–424, July 2002.

[17] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," In Proceedings of ACL, vol.10, pp.79–86, July 2002.

[18] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," In Proceedings of ACL, vol.1, pp.151–160, 2011.

[19] Y. LeCun, Y. Bengio, and G. Hinton, " Deep learning," Nature, vol.521, no.436–444, May 2015.

[20] R. Socher, C.C. Lin, C. Manning, and A.Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," In Proceedings of ICML, pp.129–136, June 2011.

[21] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," In Proceedings of EMNLP, pp.1631–1642, October 2013.

[22] R. Socher, B. Huval, C.D. Manning, and A.Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," In Proceedings of EMNLP, pp.1201–1211, July 2012.

[23] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," Neural computation, vol.9, no.8, pp.1735–1780, November 1997.

[24] K.S. Tai, R. Socher, and C.D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," In Proceedings of ACL, pp.1556–1566, July 2015.

[25] X. Zhu, P. Sobhani, and H. Guo, "Long short-term memory over tree structures," In Proceedings of ICML, pp.1604–1612, July 2015.

[26] Y. Kim, “Convolutional neural networks for sentence classification,” In Proceedings of EMNLP, pp.1746–1751, October 2014.

[27] C. N. dos Santos, and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts,” In Proceedings of COLING, pp.69–78, August 2014.

[28] Y. Zhang, and B. Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification,” arXiv preprint arXiv:1510.03820, pp.655–665, October 2015.

[29] S. Rosenthal, N. Farra, and P. Nakov, “Semeval-2017 task 4: Sentiment analysis in twitter,” In Proceedings of SemEval, pp.502–518, August 2017.

[30] M. Cliche, “BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs,” In Proceedings of SemEval, pp.572–579, August 2017.

[31] B. Dhingra, Z. Zhou, D. Fitzpatrick, M. Muehl, and W.W. Cohen, “Tweet2vec: Character-based distributed representations for social media,” In Proceedings of ACL, pp.269–274, August 2016.

[32] H. Murao, N. Kawaguchi, S. Matsubara, Y. Yamaguchi, and Y. Inagaki, “Example-based spoken dialogue system using WOZ system log,” In Proceeding of SIGDIAL, pp.140–148, August 2003.

[33] R.E. Banchs, and H. Li, “Iris: a chat-oriented dialogue system based on the vector space model,” In Proceedings of ACL, pp.37–42, July 2012.

[34] 水上雅博，N. Lasguido，木付英士，野村敏男，G. Neubig，吉野幸一郎，S. Sakti，戸田智基，中村哲，“快適度推定に基づく用例ベース対話システム，” 人工知能学会論文誌，vol.31，no.1，March 2016.

[35] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” In Proceedings of EMNLP, pp.1724–1734, October 2014.

[36] O. Vinyals, and Q. Le, “A neural conversational model,” In Proceeding of ICML Deep Learning Workshop, vol.37, July 2015.

[37] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, “A persona-based neural conversation model,” In Proceedings of ACL, pp.994–1003, August 2016.

[38] 村上聡一朗，笹野遼平，高村大也，奥村学，“数値予報マップからの天気予報コメントの自動生成,” 言語処理学会 第 23 回年次大会，March 2017.

[39] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation,” arXiv preprint arXiv:1609.08144, October 2016.

[40] 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,” Transactions of the Association for Computational Linguistics, vol.5, pp.339–351, October 2017.

[41] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.

[42] M.T. Luong, H. Pham, and C.D. Manning, “Effective approaches to attention-based neural machine translation,” In Proceedings of EMNLP, pp.1412–1421, August 2015.

[43] R. Weng, S. Huang, Z. Zheng, X. Dai, and J. Chen, “Neural machine translation with word predictions,” In Proceedings of EMNLP, pp.136–145, September 2017.

[44] A. Eriguchi, K. Hashimoto, and Y. Tsuruoka, “Tree-to-sequence attentional neural machine translation,” In Proceedings of ACL, pp.823–833, August 2016.

[45] M.T. Luong, Q.V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, “Multi-task sequence to sequence learning,” In Proceeding of ICLR, May 2016.

[46] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, “Emotional chatting machine: Emotional conversation generation with internal and external memory,” arXiv preprint arXiv:1704.01074, September 2017.

[47] M. Qiu, F.L. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang, and W. Chu, “Alime chat: A sequence to sequence and rerank based chatbot engine,” In Proceedings of ACL, vol.2, pp.498–503, July 2017.

[48] Y. Shao, S. Gouws, D. Britz, A. Goldie, B. Strope, and R. Kurzweil, “Generating high-quality and informative conversation responses with sequence-to-sequence models,” In Proceedings of EMNLP, pp.2200–2209, September 2017.

[49] L. Yao, Y. Zhang, Y. Feng, D. Zhao, and R. Yan, “Towards implicit content-introducing for generative short-text conversation systems,” In Proceedings of EMNLP, pp.2190–2199, September 2017.

[50] 赤間怜奈，稲田和明，小林颯介，佐藤祥多，乾健太郎，“転移学習を用いた対話応答のスタイル制御,” 言語処理学会 第 23 回年次大会，March 2017.

[51] 佐藤翔悦，吉永直樹，豊田正史，喜連川優，“非明示的な発話状況を考慮したニューラル対話モデルの検討,” In Proceeding of JSAI，May 2017.

[52] T. Hasegawa, N. Kaji, N. Yoshinaga, and M. Toyoda, “Predicting and eliciting addressee’s emotion in online dialogue,” In Proceedings of ACL, pp.964–972, August 2013.

[53] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” In Proceeding of NIPS Workshop on Deep Learning, December 2014.

[54] P.T. De Boer, D.P. Kroese, S. Mannor, and R.Y. Rubinstein, “A tutorial on the cross-entropy method,” Annals of operations research, vol.134, no.1, pp.19–67, February 2005.

[55] L. Xu, H. Lin, Y. Pan, H. Ren, and J. Chen, “Constructing the affective lexicon ontology,” Journal of the China Society for Scientific and Technical Information, vol.27, no.2, pp.180 –185, April 2008.

[56] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, “ BLEU: a method for automatic evaluation of machine translation,” In Proceeding of ACL, pp.311–318, July 2002.

[57] 小林のぞみ，乾健太郎，松本裕治，立石健二，福島俊一，“意見抽出のための評価表現の収集,” 自然言語処理，vol.12，no.3，pp.203–222，March 2005.

[58] H. Takamura, T. Inui, and M. Okumura, “Extracting semantic orientations of words using spin model,” In Proceedings of ACL, pp.133–140, June 2005.

[59] T. Sato, T. Hashimoto, and M. Okumura, “Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval (in japanese),” In Proceedings of the Twenty-three Annual Meeting of the Association for Natural Language Processing, The Association for Natural Language Processing, 2017.

[60] T. Sato, T. Hashimoto, and M. Okumura, "Operation of a word segmentation dictionary generation system called neologd (in japanese)," Information Processing Society of Japan, Special Interest Group on Natural Language Processing (IPSJ-SIGNL), Information Processing Society of Japan, 2016.

[61] D. Kingma, and J. Ba, "Adam: A method for stochastic optimization," In Proceeding of ICLR, May 2015.

[62] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," Journal of machine learning research, vol.15, no.1, pp.1929–1958, June 2014.

[63] J.L. Fleiss, and J. Cohen, "The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability," Educational and psychological measurement, vol.33, no.3, pp.613–619, October 1973.

[64] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, September 2016.

付録 A: Word2vec について

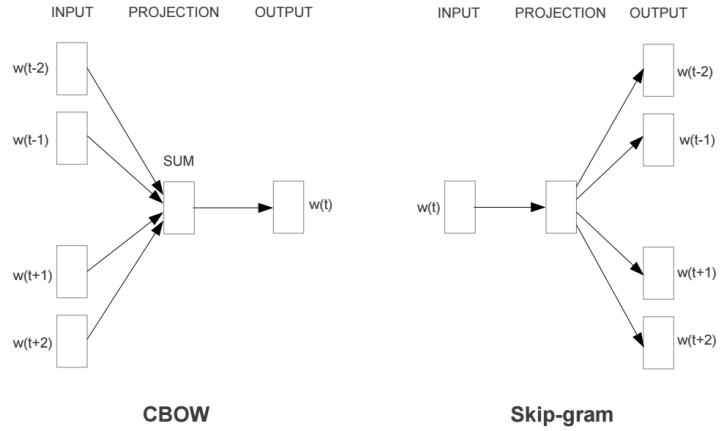

図 7.1: CBOW モデルと Skip-gram モデル（[11] より引用）

word2vec はテキストデータから単語の生起を学習するニューラルネットワークを用いた言語モデルの一つである．学習された出力データは各単語の低次元密ベクトルであり，各ベクトルが単語概念を表現しており距離計算や，今まで困難とされてきた単語間の演算も可能となった．本提案手法では，感情語彙の選定と seq2seq の入力として用いられる Embedding の事前学習に word2vec を適用している．以下に，word2vec の詳細を記述する．

word2vec の学習モデルとして, CBOW (Continuous Bag-Of-Words) モデルと Skip-gram モデルの二つが提案されている．図 7.1 において，左のモデルが CBOW であり，周辺の文脈 (周辺単語) から単語を予測するモデルである．逆に，右の Skip-gram はある単語から周辺の単語を予測するモデルである．どちらのモデルでも，入力層と中間層間の重み行列 (各単語に対する重みベクトルの集合) が，word2vec が最終的に生成する単語ベクトルの集合 (分散表現) となる．この単語ベクトルは，類似している単語がベクトル空間において近い位置にマッピングされるように表現されている．興味深い事にこのベクトルは，線形演算にも意味を含蓄しており，例えば “$king$” − “man” + “$woman$” = “$queen$” となるような結果も提出されている [10][13].

具体的な計算方法を Skip-gram を例に述べる．Skip-gram のトレーニング目的は, 周辺語の予測に利用出来る単語表現を見つけることである．つまり, 訓練用の単語列 $w_1, w_2, \ldots, w_T$ が与えられた時，Skip-gram の最終目標は式 (7) の対数確率を最大化する事に帰着される．

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

c は文脈窓のサイズ (観測する周辺の距離) であり，トレーニング精度と計算量のトレー

ドオフの関係となっている．データによるが，一般的に5〜10程度で定義される [11][12]．式 (7) で定式化されている $p(w_{t+j}|w_t)$ は softmax 関数を用いて以下のように表される．

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^{\mathrm{T}} v_{w_I})}{\sum_{w=1}^{W} \exp({v'_w}^{\mathrm{T}} v_{w_I})}$$

v_w と v'_w はそれぞれ，w の入力ベクトルと出力ベクトルである．W は辞書内の単語総数である．式 (7) は w_I から w_O が出現する確率を，全単語中から対象単語を選択するような softmax 関数で定義している．この関数の分母に注目すると全単語分の和を計算するので，この勾配計算の計算コストが跳ね上がってしまう．この計算オーダーを減らす方法として，階層的 softmax と負例サンプリングが存在する．

階層的 softmax は単語の出現の代わりに意味を使用したモデルであり，具体的には単語をクラスタリングし，バイナリの2分木を生成し，各単語の出現確率を根 (root) からその単語までの各ノードのベクトルと内積を取り，その値を入力値としたシグモイド関数の積で単語の生起確率を近似する方法である．具体的な計算式を以下に記す．

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma([n(w,j+1) = ch(n(w,j))] \cdot {v'_{n(w,j)}}^{\mathrm{T}} v_{w_I})$$

$n(w,j+1)$ は，根から w までの道における j 番目のノードであり，$L(w)$ はこの道の距離である．従って，$n(w,1) = root$，$n(w,L(w)) = w$ である．任意の内部ノード n に対して，任意の修正された n の子ノードを $ch(n)$ としている．$[x]$ は x が真であれば1，偽であれば -1 と定義する．また，$\sigma(x) = \frac{1}{(1+\exp(-x))}$ である．$\sum_{w=1}^{W} p(w|w_I) = 1$ であることから，$p(w_O|w_I)$ の計算量は $L(w_O)$ に比例し，計算量の平均は $\log W$ を超えない [12]．実際の木の生成方法としてハフマン木が挙げられる．単語ごとにハフマン符号を割り当てる際に，頻出単語に短い符号を割り当てる事で高速アクセスを可能にしている．

負例サンプリングは，式 (7) の分母の全単語に関して和をとる代わりに，単語分布に関する期待値計算に近似させる手法である．期待値計算に関しては，複数個のサンプルを取って計算する．実際には，5〜20個程度のサンプルで良いとの結果が出ており [12]，全単語数のループがわずか数回のループで終わるというメリットがある．計算式を以下に挙げる．

$$\log P(w_O|w_I) = \log \sigma({v'_{w_O}}^{\mathrm{T}} v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \{\log \sigma(-{v'_{w_i}}^{\mathrm{T}} v_{w_I})\}$$

σ 内の期待値計算は，式の通り k 個 (5〜20) の負例サンプルを抽出して近似させる．この計算はロジスティック回帰分布を用いてノイズ分布 $P_n(w)$ から抽出された w_O を区別する

意味合いを含んでいる．$P_n(w)$ は 1-gram 頻度分布の $\frac{3}{4}$ 乗に比例させた時がもっとも性能が高いことが報告されている [12].

モデルは確率的勾配降下法 (SGD : Stochastic Gradient Descent) [64] を用いて最適化され，ベクトルと重みベクトルの更新を単語毎に更新する．

付録B: バリデーションセットによる BLEU 及び WER の評価値の推移

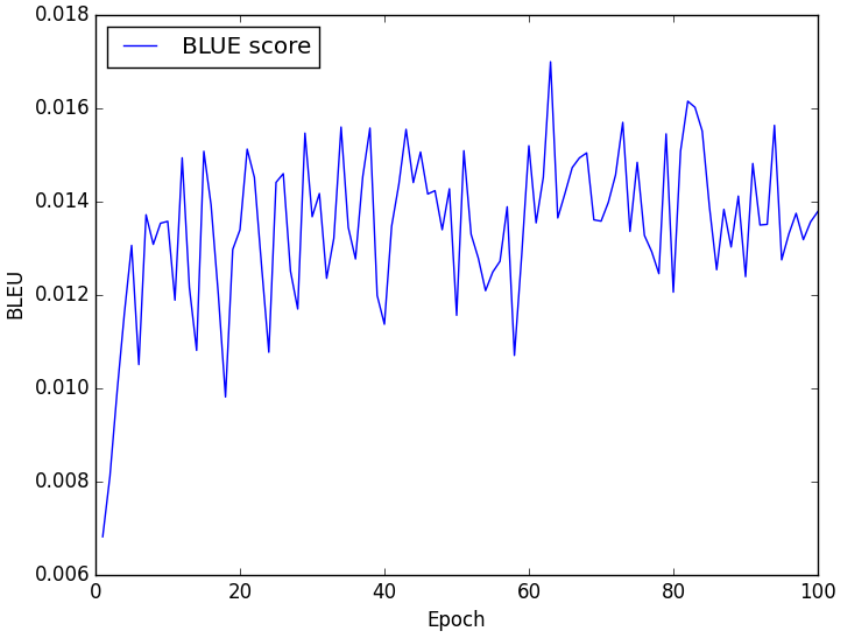

図 7.2: 既存手法のエポック毎の BLEU 値

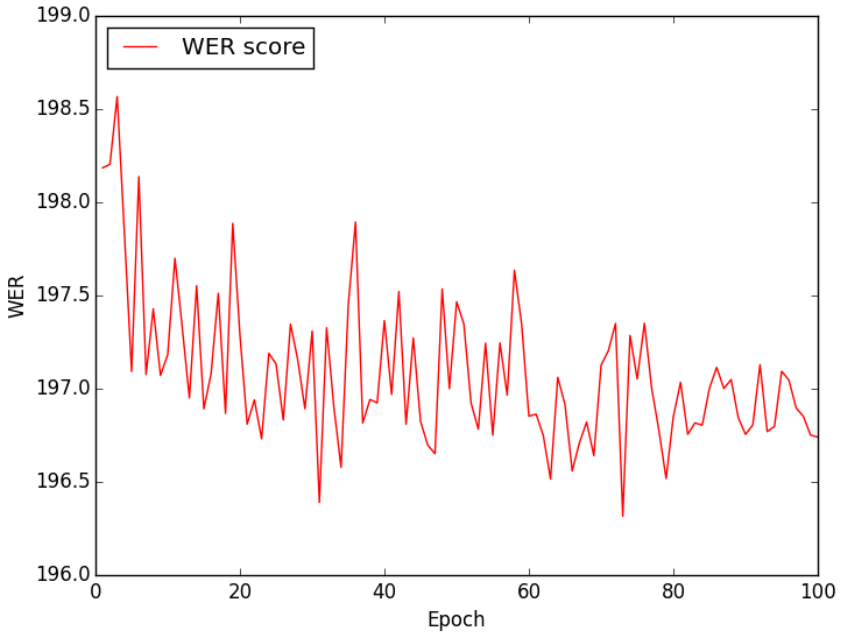

図 7.3: 既存手法のエポック毎の WER 値

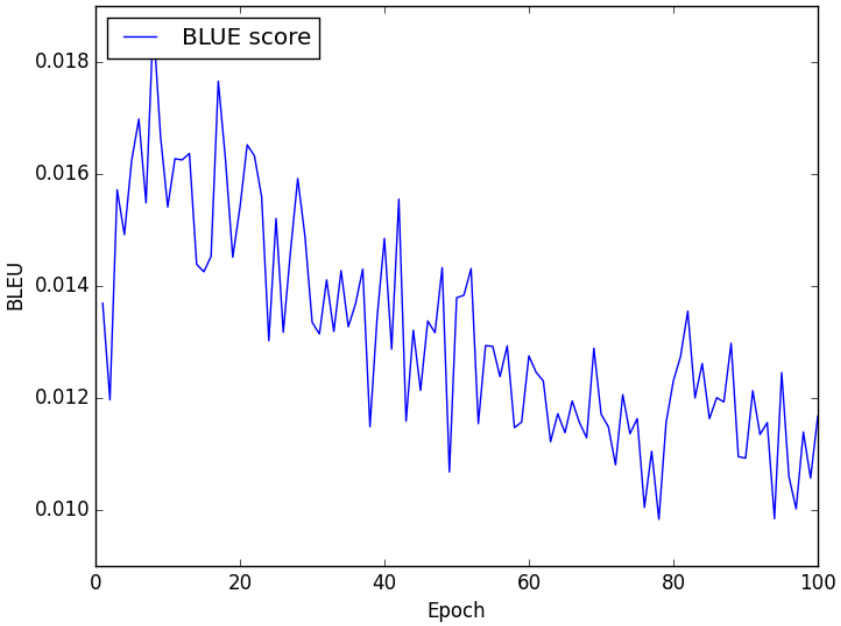

図 7.4: 提案手法のエポック毎の BLEU 値

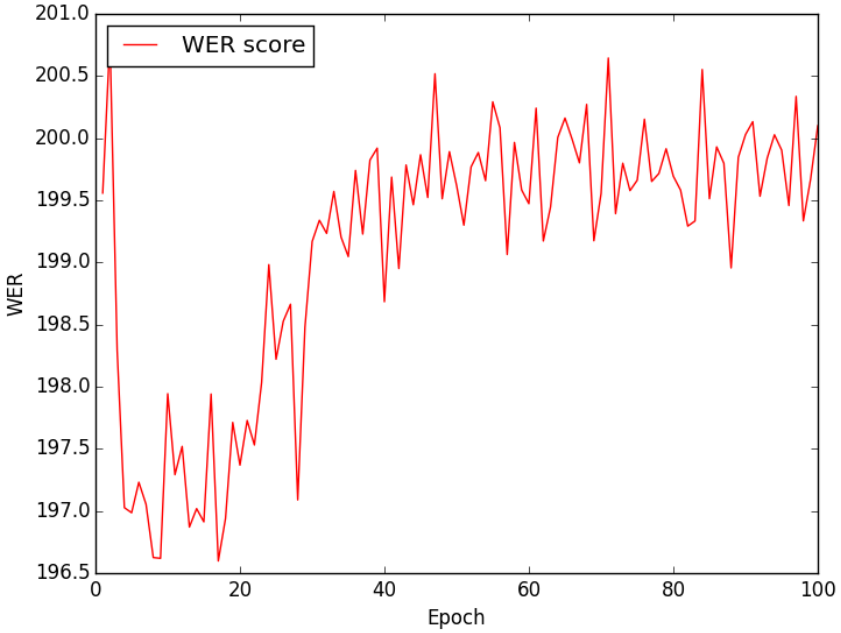

図 7.5: 提案手法のエポック毎の WER 値