

Contents

1	序論	1
1.1	研究背景及び目的	1
1.2	本論文の構成	2
2	対話システムの関連研究	2
3	想定する対話システムの全体像	3
4	日本語データの取り扱いについて	4
4.1	調査) 発話データ	4
4.1.1	フィルタ	4
4.1.2	調査結果	4
4.1.3	考察	6
4.2	調査) 対話データ	7
4.2.1	調査結果	7
4.2.2	考察	8
4.3	問題設定	8
4.3.1	関連研究	9
4.4	実験) 漢字かな問題に対する単語分散取得	9
4.4.1	実験概要	9
4.4.2	実験結果	9
4.4.3	考察	9
4.5	実験) 得られた単語分散を用いた極性判定	9
4.5.1	実験概要	9
4.5.2	実験結果	9
4.5.3	考察	9
5	文抽出を念頭においた不均衡分散・サイズの分類問題	9
5.1	問題設定	9
5.2	実験) 自然言語処理の場合における一般的なクラス分類	10
5.3	実験) 画像タスクに置換した場合における一般的なクラス分類	10
5.4	実験) 自然言語処理の場合における点類似度を用いたクラス分類	10
5.5	実験) 画像タスクに置換した場合における点類似度を用いたクラス分類	10
5.6	考察	10
6	機械翻訳システムを用いた対話モデル	10
6.1	問題設定	10
6.2	実験) Seq2Seq Attention と Transformer の精度比較	10
6.2.1	実験概要	10
6.2.2	実験結果	10
6.2.3	考察	10
7	文のスタイル変換	10
7.1	関連研究	10
7.2	問題設定	10
7.3	実験) 書き言葉→話し言葉のスタイル変換	11
7.3.1	実験概要	11
7.3.2	実験結果	11
7.3.3	考察	11

8	CoLA タスクを応用した対話システムのエラー検知	11
8.1	問題設定	11
8.2	実験) 対話システムのエラー検知	11
8.2.1	実験概要	11
8.2.2	実験結果	11
8.2.3	考察	11
9	付録	11
9.1	対話システムの関連研究	11
9.1.1	Sounding Board	11
9.1.2	Gunrock	11
9.2	日本語データの取り扱いについて	11
9.2.1	単語分割	11
9.2.2	形態素解析	12
9.2.3	NER	12
9.2.4	Word Piece	12
9.2.5	Sentence Pieces	12
9.2.6	Skipgram	12
9.2.7	CNN-RNN	12
9.3	質問文抽出を念頭においた不均衡分散・サイズの分類問題	12
9.3.1	画像データ	12
9.3.2	文データ	12
9.4	機械翻訳システムを用いた対話	12
9.4.1	Seq2Seq Attention	12
9.4.2	Transformer	12
9.5	文のスタイル変換	12
9.5.1	Sequence to Better Sequence	12
9.5.2	CopyNet	12
9.5.3	Denoising Auto Encoder	12
9.6	CoLA タスクを応用した対話システムのエラー検知	12
9.6.1	BERT	12
10	結論	13
10.1	今後の課題	13
11	C++ コードを Emacs + Org mode で書く	14

List of Figures

1	りんなのフレームワーク	2
2	本研究のシステム全体像	3

1 序論

1.1 研究背景及び目的

ある目的に対してより完璧に (accuracy が高くなるように) 命令を実行をする Artificial Intelligence が求められている昨今の AI 競争の時代に対し、自然言語処理やゲーム AI のようなタスクは極めて複雑な課題を抱えている。例えばそれは“言葉”という問題である。これは人間がコンピュータに正解となるものを提供することが極めて難しく、“なんとなく良い感じに”目的を達成してくれることを期待することが多い。この問題に対処するための手段として、入手でき得る限りの大規模なデータを用意して中心極限定理的に尤もらしい中心部を得る方法や、とにかく何らかの単一のモデルに押し込めて問題を解くという方法¹がある。それに対して、データそのものを一旦精査・前処理すること、問題を整理・分解しそれぞれを解くことも研究²として存在している。

自然言語処理の、特に対話システムについて考えたとき、小問題に分割した上で対話システムを達成した例として、例として Amazon Alexa Prize³ というコンテストや Microsoft 社が研究・開発している“りんな”⁴を挙げることができる。これらは対話を行うという問題に対して小さな部分問題を解くタスクを設定し、それぞれを組み合わせることで元の問題を解くというスタイルを取っている。

本研究ではこれらを参考に、日本語の対話システムを作成するという問題に対して小問題を設定しそれを解くための手法を提案・実験する。またその前準備としてデータ収集に絡めて日本語データとその前処理について考察する。尚本研究が最終的に望むものは、キャラクター性を持った対話可能なエージェントを作ることであることを強調する。

もう少し言及すると、本研究ではデータからモデルにかけて5つの少テーマについて研究を行った。概要をそれぞれ説明すると以下ようになる。

1. 日本語データの取り扱いについて

我々は一般に日本語を話しており、それを用いた対話システムの構築が本研究の主目的である。しかし機械学習等のデータセットや実験で多く使われているのは、日本語とは使っている文字や文型で大きく異なっている、英語のことが多い。その前提のもとで日本語のデータ、特にセンテンスに対して、どのような性質があるのかを調査し、また提案する漢字→ひらがな変換という前処理とそれによって得られる性質についても議論を行う。

2. 文抽出を念頭においた不均衡分散・サイズの分類問題

テキストのカテゴリ分類を考えたとき、一般には n 個のカテゴリの中から任意の文が入力されることを想定している。今回はそれとはやや問題設定が異なり、いくつかの文をカテゴリ $1 \dots n-1$ 、それ以外をカテゴリ n として扱うことを主問題とした。しかしデータを設定・収集することが困難であったため、一部画像認識の問題として議論を行う。

3. 機械翻訳システムを用いた対話モデル

一対一対話を行う際に、機械翻訳システムを用いることがある。今回はそれを、問題を文脈に依存しない発話に対する反応を学習することに再設定し、Transformer という 2017 年 12 月時点で SOTA (State-Of-The-Art 最高水準) を獲得した機械翻訳の手法を用いて実験、有名な手法である Sequence to Sequence Attention を用いた一対一対話モデルと比較を行う。

4. 文のスタイル変換

文のスタイルとは、例えば口調や訛り、書き言葉や話し言葉といったものを指す。これは日本語で特に顕著に見られるもので、テキスト上でもこれを確認することで相手のペルソナをある程度想定することができる。本研究が日本語を対象としていること、キャラクター性を持たせたいというモチベーションがあることから文に特定のスタイルを持たせることを問題として取り上げる。

5. CoLA タスクを応用した対話システムのエラー検知

対話システムを小問題に分割して解く弊害として、それぞれの問題でエラー (不適切な出力) が出てしまうというものがある。これに対処するため、特に何らかのモデルから生成された文に対しそれが自然であるかどうかを評価するモデルを作成し実験する。

¹HRED (Sordani et al. 2015) や VHRED (Serban et al. 2016) があるが、発話の多様性を得ること (一般的な受け答えを学んでしまい、同じような文ばかり生成してしまう) やデータを十分に集めることが難しいなど課題がある。

²日本で人気を得ている“マルチモーダルエージェント AI”とは、複数のソースから問題を見直すという特徴があるが、これは複数のモデルを使っているという意味ではあるが、問題を分割しようとしているわけではないという点でこの研究と大きく違うと言えるだろう。

³<https://developer.amazon.com/alexaprize>

⁴https://twitter.com/ms_rinna

1.2 本論文の構成

第1章に本論文の概要とその構成について説明を行い、第2章で関連研究を紹介し、第3章で本研究で掲げるシステムの全体像を示す。そして第4章から第9章にかけては1.1で述べたテーマについての順に議論する。その後付録として補足をまとめたものを第10章として示す。最後に議論として本論文のまとめ、今後の展望について述べる。

2 対話システムの関連研究

対話システムの関連研究としては、1.1で述べたように Amazon Alexa Prize というコンテストや、Microsoft 社のりんなを挙げることができるだろう。Microsoft 社のりんなは日本語雑談対話 (Wo et al. 2016) を実現しており、2018 年現在 Twitter などでは活動をしている。Amazon Alexa Prize は Amazon Alexa という音声会話を行うことのできる端末に搭載する対話システムを競う大会である。評価対象はユーザの印象であり、別の指標として対話時間が公開される。2018 年度の Amazon Alexa Prize では平均 10 分程度の対話を行うことの出来たシステム⁵が優勝した。顔や体といったテキスト以外の情報を用いることの出来ない対話システムでこのような結果が得られたことは注目すべきことである。

いずれも複数のモデルを組み合わせで構成されており、例えば言語理解部と文生成部、そして本研究で取り扱わないものとしては、音声理解部と音声生成部を挙げることができる。またりんなに関してはそれに加えて画像認識部などの対話以外の⁶システムも構築している [Figure 1]。

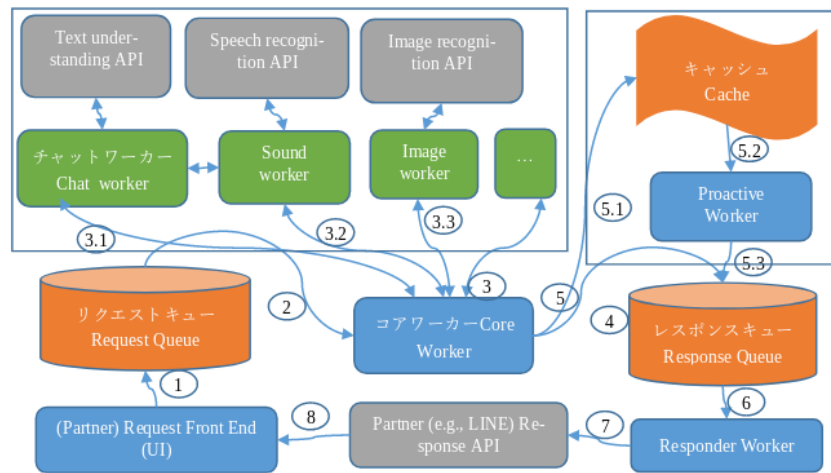


図 2. りんなのフレームワーク

Figure 1: りんなのフレームワーク

⁵2018 年度優勝は カルフォルニア大学デビス校のチームが開発したの Gunrock というシステムであり、また 2017 年度優勝はワシントン大学のチームが開発した Sounding Board というシステムである。この 2 つについての詳細は 9.1 で紹介する。なぜこれらを追実装しなかったのかという疑問もあるかもしれないが、いずれも大規模なデータを必要とする (例えば 10M を超える会話データ) ため、個人でそれを実装することは不可能である。

⁶対話をテキストやそれを示す音声のみのコミュニケーションと定義した場合。実際には対話には身振り手振り、表情といった要素が複雑に絡んでいる。そのため 2017 年頃からは、表情を考慮した対話システムが提案され (Chu, Li, and Fidler 2018) 研究されている。

3 想定する対話システムの全体像

以下に本研究で想定する対話システムの全体像を示す [Figure 2]。

このシステムでは入力としてテキストと、環境情報を得る。このシステムにおける環境情報とはこのシステムが組み込まれているエージェントが居る場所の環境 (天候や気温・湿度)、エージェントの内部状態 (メモリ使用率等) を指す。これはテキストを用いた人対人の対話をイメージしたもので、つまり相手の居る環境、相手の体調をそれぞれ置き換えたものになる。また Answer Generation に用いる所謂個人データのようなものもエージェントの内部に持っているものとする。本論文で扱うものは、この内の Sentence Detection / Sentence Categorization / Topic Dialogue / Style Transfer である。また Topic Dialogue から Style Transfer への矢印・Answer Generation から Style Transfer への矢印・Style Transfer から Output への矢印におけるエラー検知についても議論する。

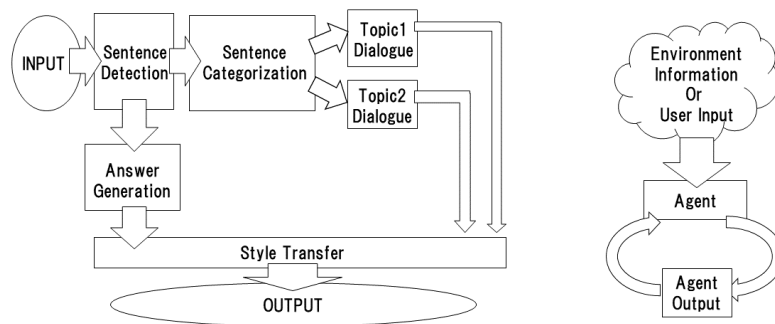


Figure 2: 本研究のシステム全体像

- **Sentence Detection** [該当部:文抽出を念頭においた不均衡分散・サイズの分類問題]
ある特定の文を取り出す。取り出された場合はどの意味として取り出されたのかという情報とともに、Answer Generation へ向かい、取り出されなかった場合には付加情報なしで Sentence Categorization へ入力を受け流す。最終的にはほとんどの文をここで抽出し、それに対する返答を Answer Generation でエージェントの内部状態ないし外部知識ベースを参照しながら生成する。
- **Sentence Categorization** [該当部:日本語データの取り扱いについて]
文を大雑把にカテゴリ分類する。例えばそれは livedoor news corpus⁷ で議論されるような スポーツ/IT/家電 といったようなカテゴリである。ここでカテゴリ分類された文はそれぞれ対応する Topic Dialogue に流される。
- **Topic Dialogue** [該当部:機械翻訳システムを用いた対話モデル]
与えられたカテゴリに対する一対一応答を行う。例えばゲームについての話題を受け持つ Topic Dialogue はゲームに関する入力文を期待しており、それに対する出力を学習しているものとする。そのモデルはエージェントのペルソナに応じて置換することが可能であり、例えば好きなゲームカテゴリについての好意的なデータを多分に含んだデータセットで訓練した Topic Dialogue はそのゲームカテゴリが好きな (好きになった) エージェントが持つことになる。
- **Style Transfer** [該当部:文のスタイル変換]
文のスタイルを変換する。ここで言う文のスタイルとは例えば書き言葉や話し言葉、各ペルソナに基づいた語尾変化を示す。
- **エラー検知についての議論** [該当部:CoLA タスクを応用した対話システムのエラー検知]
上記のシステムで発生するエラーデータと正常なデータを分類する。

⁷<https://www.roundhuit.com/download.html#ldcc>

4 日本語データの取り扱いについて

日本語データは英語データに比べていくつかの問題を抱えている。問題の例としては、文字の数が多すぎること、スペースといった意味ごとの分割がないこと、容易にペルソナを特定できるような多彩な語尾変化があること、多国語も日本語であるかのように用いること、同意同音の語でも様々な表記方法があることが挙げられる⁸。

また一般に公開されている対話データセットを対話テキストのみで学習させると想定したとき、背景知識の欠如を指摘せざるを得ない。更に言えば日本人の特徴として“言外にわかり合う”というコミュニケーションスタイルも問題を難しくしていると言えるだろう。

この章では上記の問題があることを公開されているデータセット、Twitter から収集したデータセットを用いて調査するとともに、“漢字をかなに変換する”という前処理を用いることでどのようにデータの性質が変化するかを、単語分散を得るというタスクについて実験する。

尚本研究では、形態素解析には MeCab 0.996、単語辞書として mecab-ipadic-neologd 20181112-01 を用いた。特に Twitter のようなデータは流行語や新語に対応するため、単語辞書を定期的に更新する必要がある。

4.1 調査) 発話データ

発話データとして、2018 年 12 月 25 日 23:00 頃 から翌 26 日 10:00 頃 までに収集した 7 万件の Twitter データを収集し、その性質を観測した。

データの収集手法としては Twitter 社が公開している API を用い、日本のユーザから呟かれている内容を集めるものとした。この処理によって生データが 77,285 発話得られた。

4.1.1 フィルタ

データを収集するにあたり、タグや宛名、URL リンクと言った Twitter に特有な部分を省いた。その上で、4 文字以上、60 文字以下のデータをすべて抽出し、データを 54,368 発話にした。

Twitter に特有な部分を省いた理由として、全体の目的から考えて Twitter データに特化させる必要がなかったこと、タグは時系列で発生・消滅すること、宛名に関してはそのユーザの背景情報が必要になることが容易に想像できること、URL リンクを発話として認めるべきではないと考えたこと⁹を挙げる。

また文字数でフィルタを行った理由として、1. 4 文字未満のデータは少なく、この後議論する単語分割が出来ないようなデータ、そのみでは意味が通じないデータが多く含まれていたこと、2. 60 字超過のデータは何らかの内容に対する説明と言った発話データとはややベクトルの異なるデータが多かったこと、深層学習を中心とした機械学習を用いた自然言語処理(要約タスクを除く)に用いるデータであると考えたとき、長すぎるテキストは短くされる前処理を施すことが一般的であること、を挙げる。

Table 1: 発話データに対して適用したフィルタとその理由

フィルタの概要	詳細	理由
Twitter 特有の内容	タグ 宛名 URL リンク	時系列で発生・消滅するため 宛名のユーザに対する情報が必要であるため リンクを発話として認めるべきか議論の余地があるため
文字数	4 文字未満 60 文字超過	データ数が少なかったため 単語分割が出来ないため(極端な略語など) 発話データというよりは説明のようなデータが多かったため 適用する予定の手法では情報の一部が切り落とされてしまうため

4.1.2 調査結果

フィルタによって抽出された 54,368 発話を調査した。

まず発話データとして問題があると考えられる発話について報告する¹⁰。

⁸前 2 つに関しては、中国語も共通して抱えている問題と言える。

⁹勿論タグに意味が込められている例(“#〇〇を許すな”など)も多く見られたが、タグを認めるとタグのあるすべてのデータを手動で確認する必要があったため今回はすべて省いた。

¹⁰すべての報告における例は、個人情報を含んだ部分を含まないように編集されている。

Table 2: 発話データの調査結果 1

概要	詳細	例
他国語を用いた発話	中国語・英語等を用いた(含まれる) ツイートが 0.5 % 程度見られた	Very nice Merry Christmas! 謝謝 Guten Morgen!
テキストのみでは 理解できない発話	画像などのコンテンツに 対する発話が微量見られた ハイコンテキスト過ぎて 理解できないものが見られた	これ最高 れ!!!
(意図的・意図的でない) 誤字		オフトウン イケメソ
顔文字や絵文字の多用	Twitter で許可されている絵文字や、 顔文字が含まれる発話が 8% 程見られた	(*´ω`*) お疲れ様です [(:3[■ ■]] (´▽`)>
単語の一部や 語尾の繰り返し	特に感情的なつぶやきでは、 強調などの目的から 語の一部を繰り返す傾向が見られた	全全全休 ほにゃほにゃほにゃほにゃする やだあああああああ!
略語の多用	長い単語、文は相互に理解できるような 形に省略されることが多かった	メリクリ! なるはや
別の表現	同じ意味を示すが 別の表記法があるものは 共通化されているわけではなかった	!/!!!!/!/!!/!!!!!!!!!! ... /... こんど/今度 彼氏/カレ氏/カレシ デス/です
伏せ字	隠語など伏せ字を用いている場合があった	○ね
語尾の特徴付け等		ねれないぽよ ...と思うニョロ むいねー

次に主に情報の価値として問題があると考えられる発話について報告する。

Table 3: 発話データの調査結果 2

概要	詳細	例
個人情報の入ったもの	電話番号や SNS の ID などを 含まれるものが、 一万件に対して 5,6 件あった 個人名・アカウント名が含まれるものを 含めると 5%程になってしまった	
時刻など		2018.12.26 06:00
頻度が高すぎるもの	挨拶等	メリクリ! おはよう
センシティブなもの		
Twitter 特有のもの		凍結された フォローありがとうございます
数値データ	英語での NLP の一部では積極的に削除されている 漢数字 ギリシャ数字	2018 200 円 一 V

最後にこの後実験として取り上げる極性判定のデータとして問題があると考えられる発話について報告する。

Table 4: 発話データの調査結果 3

概要	詳細	例
予定などのメモ書き	個人の予定やイベントの告知	
企業などの広告		
取引などのツイート		買) 鳥獣戯画のペンダント
豆知識や引用	特に深夜～早朝にかけては自動ツイートのような形式の豆知識や引用の頻度が高くなっていた最大では3%程がこれに含まれていた	丁字染ちょうじぞめ オロバス ¥n ソロモン 72 柱の… [飲み会で使える? ダジャレ]… サーッ!(迫真)
感情が含まれているか 疑問のあるデータ		なぜ僕らは生きるのか

4.1.3 考察

データを収集した時間も相まって広告や豆知識・引用といった発話が多く観測された。これらのデータは極性判定やカテゴリ分類、ユーザクラスタリングなどに悪影響を与えることが論理的に考えられる。予定や広告、時刻などに関係したデータは、ほとんどの場合で一過性のものであるため長期的なシステムのためのデータとして見たときには適切であるか疑問が残る。

数値データや個人名のようなデータに関しては、英語での NLP、特に良い精度を持ったいくつかのタスクに対しては何らかの記号に置換されることが多い。しかし日本語でこれを適用しようとしたとき、1. 様々な表記方法があること、2. スペースで分割されていないため、形態素解析などの技術や NER(Named Entity Recognition 固有表現抽出) の技術を組み合わせなければ抽出できないこと、が問題として挙げられる。特に形態素解析に関しては Twitter のデータのような正規化されていないテキストに行った場合、精度が比較的に落ちるため、何らかの精度向上手法または別手法を提案する必要がある。

また同じ意味を表す文でも様々なバリエーションがあることがわかった。例えば“おはよう”を例に取ってみると、“おはようございます”、“おはよう”、“おは”、“おはようお”、“おは(愛称等)”といったバリエーションが見られた。これらはキャラクター性を持たせるためには必要な分散であるが、意味のみに注目した場合や、語彙数の問題を考慮した場合には極力減らされたほうが良いと考えられる¹¹。更にバリエーションのある文は平均的に出現頻度が高い¹²ため、これを集めすぎるとデータに偏りが生まれてしまうことも考慮する必要があるだろう。

極性判定のみに絞った議論をするならば、例えば自動ツイートされた発話にはユーザの極性があるとは考えにくいので、これを省くのが適当であると考えられる。しかし以上のことを踏まえてデータの再抽出・編集をフィルタリング後のデータの中の、15,000 程度のデータに対して行ったところ、1,500 程度のデータしか得られなかった¹³。

形態素解析で成功した例

りかちゃんありがとう

<形態素解析結果>

りか 名詞, 固有名詞, 人名, 名, *, *, りか, リカ, リカ

ちゃん 名詞, 接尾, 人名, *, *, *, ちゃん, チャン, チャン

ありがとう 感動詞, *, *, *, *, *, ありがとう, アリガトウ, アリガトー

¹¹ 英語の NLP (例えば機械翻訳) でも前処理として、“he’s”を“he is”にするなどの前処理が行われることがある。

¹² 例えば 26 日午前 6 時ちょうど頃は 3 割程度が宛先や顔文字などの付加情報の差はあれど“おはよう”の意味の発話であった。

¹³ これは公開できる形にするため、個人情報を含んだデータを編集・削除したこと、極性を持たないと思われるデータ(中性という意味ではない)を省いたことが大きく起因している。

形態素解析で失敗した例

山さんに …

<形態素解析結果>

山 名詞, 一般,*,*,*, 山, ヤマ, ヤマ
 さん 名詞, 接尾, 人名,*,*,*, さん, サン, サン
 に 助詞, 格助詞, 一般,*,*,*, に, ニ, ニ
 …

※人名を指すが一般名詞として認識されてしまっている。
 このよう場合には単語分割した後、NER を用いて検出することが望ましいと言える。

4.2 調査) 対話データ

対話データとして、2018 年 8 月から 12 月にかけて不定期に Twitter から収集した対話データ、一般公開されている書き起こしの対話コーパス、一般公開されているチャットの対話コーパスについてデータを観測した。

以下に調査結果として何らかの問題があると考えられる特徴について報告し、それに対する考察を述べる。

4.2.1 調査結果

1. Twitter から収集した対話データ

収集方法は Twitter 社が公開している API を用い、日本のユーザから呟かれている内容の中から、3 発話以上対話が続いているものを収集した。この処理によって生データが 10,767 の対話ペアが得られた。そして生データに対しては 4.1.1 と同様にハッシュタグと宛名、そして URL リンクを削除したが、文字制限は対話間の意味を観測するため行わなかった。

Table 5: 対話データの調査結果 1

概要	詳細	例
センシティブな内容	3 %程はセンシティブな内容の対話であった。	
ゲームに関する内容	5 %程はゲームに関する内容であった。 その中には一過性の内容 (情報共有や待ち合わせ等) が含まれていた	
顔文字や絵文字等が含まれるもの	15 %程は顔文字や絵文字を含んでいた そのうちの 2 割ほどは顔文字・絵文字のみが発話になっているものがあつた	おはようございます! ((* ° ㇿ °) ノ (' ㇿ `)
似たような内容	特に挨拶など同じような 内容の対話頻度が高かつた 朝方には半数が “おはようございます” の内容であつた	おはようございますよ
事前知識を必要とする内容	間柄や話題 (例えばゲーム) の内容に 関する事前知識がいるものが 多く感じられた。 ¹⁴	line カメラたのしい
固有表現が含まれるもの	名前等固有表現が含まれるものは 3 割程度であつた。	

2. 名大会話コーパスから収集したデータ

名大会話コーパス (逸子, 美恵子, and ディヴィッド義和 2011) から入手できる 129 会話について観測した。名大会話コーパスとは日本語母語話者同士の雑談を文字化したコーパスで、129 会話を収録、

¹⁴ アノテータが一人のため境界を判定することは難しいため、割合を明言することは出来ない。

その合計時間は 100 時間に及ぶ比較的大規模なものだ。ライセンスがクリエイティブ・コモンズ表示-非営利-改変禁止 4.0 国際ライセンスで公開されているため、研究目的で用いることが非常に容易なコーパスであると言える。

非常に大規模かつ考察で述べるように複雑な内容であるため、出現頻度については言及しない。

Table 6: 対話データの調査結果 2

概要	詳細	例
言外のコミュニケーション	言語化せずに伝える内容があった	<笑い> (共感の意)
長文や複文	相手が内容を理解したものとして文を継続させる場合があった。	すごい勢いで走って。 私、あ、あーさっきの犬だとか 私たちが言っとるじゃん。 犬も気がついたじゃん。 じゃははって走ってきちゃって、犬が。
書き言葉・話し言葉の変化	あの → あん といった変化が見られた。	ほいでさあ、ずっと歩いていたんだけど、 そうすと上から、なんか町の中が見れるじゃん。
固有表現	個人情報保護のため 名前などの 固有表現は置換されていた	***の町というのはちいちゃくって... ほいで、あの、F023 さんはあたしが前の日に... C が、あの一、写真を見せてくれたんだけど...

3. 対話破綻チャレンジ 2 から収集したデータ

対話破綻チャレンジとは人間と対話システムとの間で生じる「対話破綻」(ユーザが対話を継続できなくなる状態)を自動検出することを目的とした、評価型ワークショップである。

このデータは対話システムと人間間とのテキストを用いた対話データと、その対話が成立しているかどうかを判定した複数人によるアノテーションが含まれており、本研究の目指すエージェントと人の対話の形に最も近いデータセットであると言える。

本データセットは問題点が少なく、アノテーションに従って、比較的成立しているとみなされた対話を抽出することで対話データを生成することが出来た。

4.2.2 考察

Twitter から収集した対話データに関しては Twitter データとして非常に有効であると考えられる。しかし比較的にセンシティブな内容が多く、これを対話データとして学習させてしまうことによる、対話システムの倫理的な問題を考慮しなければならないだろう。また顔文字や絵文字等は 4.1 で考察したように単位で分割することが難しい。同様に同じような意味を持った対話が多く存在していたことから、これにも対処する必要があるだろう。

名大会話コーパスから収集したデータに関しては日常会話を分析・理解するには抽出するには非常に価値のあるデータセットであるが、これをチャットのようなテキスト入力等を介した対話には不適切なデータであると考えられる。このコーパスを観測して考察できる内容としては、1. 書き言葉・話し言葉の変化は想像以上に大きなものであったと言えること、2. 決して発話一つに対して返答が一つという形式になっているわけではないこと、3. 固有表現の取扱についてより深く考察する必要があること、であった。

対話破綻チャレンジ 2 から収集したデータはほぼ申し分ない自然さを持ったデータを集めることができた。しかし対話システムと人との対話データであるため、“人対人のような日常会話” 対話は比較的少なく、“人のような” 対話エージェントを作成するならば、不足している対話を外部から付け加える必要があると考えられる。

4.3 問題設定

NLP の研究分野の一つについて単語分散を用いた言語モデル生成がある。単語といったある単位ごとの意味をベクトルなどの数値にする手法であり、この利点としては、単位ごとの距離を考えたとき、意味的に近い要素は近く、遠い要素は遠くなることで様々な NLP のタスクで自然言語を数値化する際に、自然言語の特徴を強く表すことができるようになるというものがある。

本研究ではこの単語分散を得るという問題に対してデータの前処理がどのように影響するのかを理解する目的で、2 つの実験を行う。

一つは、1. 漢字・かな入り混じり文、2. かな飲みに変換した文、によって得られる単語分散の性質の違

いを確認する実験、もう一つは得られた単語分散を用いて極性判定を行う実験である。
TODO: Word Embedding で得られる T-SNE 画像を挿入する

4.3.1 関連研究

単語分散を得るための手法としては、SVD(特異値分解)(TODO:svd) や Word2Vec(TODO:word2vec) や glove(TODO:glove)、fasttext(TODO::fasttext) といった手法が有名だ。また昨今、NLP では文単位での解析が多いこと、文全体の意味も考慮したほうが良いというモチベーションから、単語分散のみならず、文ごとの関係も考慮してベクトルを生成する手法が提案されている。その代表例が、ELMo(TODO:ELMo)、BERT(TODO:BERT) と言った深層学習のモデルであり、昨今の様々な NLP のタスクで SOTA を達成している。

4.4 実験) 漢字かな問題に対する単語分散取得

この実験では、日本語特有に存在する“漢字とかなによる同意表現の複数表記”を解消するための漢字
→

4.4.1 実験概要

単語分散を得るためのコーパスとして Wikipedia コーパスを用いたことなど、実験の概要を示す。

4.4.2 実験結果

実験結果を示す。

4.4.3 考察

考察を示す。

4.5 実験) 得られた単語分散を用いた極性判定

4.5.1 実験概要

4.4 で得た単語分散を用いて極性判定を行ったこと、極性判定のデータセットは 4.1 で抽出・編集したデータであることを示す。(抽出・編集条件を再度示す) また実験に用いたネットワークについて説明する (CNN-RNN)

4.5.2 実験結果

実験結果を示す。

4.5.3 考察

考察を示す。

5 文抽出を念頭においた不均衡分散・サイズのカテゴリ分類問題

5.1 問題設定

入力された文が特定の意味を持った文であるかどうかを抽出する問題において、どのように分類すべきなのかを検討する。一般的なクラス分類との比較として、この問題は特定の意味を持った文の集合であるクラスと、それ以外のクラスとでデータの分散やデータの数に大きな差があること、画像認識と違ってアップサンプリング(水増し)が難しいことを問題点としてあげ、まず一般的に用いられている分類問題として解き、次に提案する手法である点類似度を用いたクラス分類を説明する。(特定の文で分岐を行い、その組み合わせを用いてユーザとの対話を試みる、シナリオ型対話システムがあることにも触れる。) 考察は比較のためにすべての実験のあとにまとめることを説明する。

5.2 実験) 自然言語処理の場合における一般的なクラス分類

news20 というデータセットを用いて CNN を用いた 1 クラス分類 (1 カテゴリ : 19 カテゴリ) を行う。相手のクラスの分散が想定よりも小さいことを注記する。

5.3 実験) 画像タスクに置換した場合における一般的なクラス分類

imagenet の画像タスクで、猫・犬分類と猫・ランダム画像でのクラス分類を行う。

5.4 実験) 自然言語処理の場合における点類似度を用いたクラス分類

BERT モデルを用いて、文類似度を測り、それを用いてクラス分類を行う。

5.5 実験) 画像タスクに置換した場合における点類似度を用いたクラス分類

画像の類似度を測り、それを用いてクラス分類を行う (実験が間に合えば)

5.6 考察

後者のほうが拡張性があること、前者の場合に猫・犬よりも猫・ランダムのほうが精度が悪くなる傾向があることを指摘する。

6 機械翻訳システムを用いた対話モデル

6.1 問題設定

反射応答のような問題について、機械翻訳を用いて発話を行わせることを提案、その手法として昨今機械翻訳の分野で SOTA を取っていた Transformer を用いることを実験し、その性能を考察する。

6.2 実験) Seq2Seq Attention と Transformer の精度比較

6.2.1 実験概要

データセットなどの実験概要を示す

6.2.2 実験結果

実験結果を示す。

6.2.3 考察

考察を示す。

7 文のスタイル変換

7.1 関連研究

この分野の関連研究として sequence to better sequence(本実験) や、(夏季レポートに記載したもの) を例に挙げる。(画像認識の分野におけるスタイル変換についても触れておく必要があれば触れておく)

7.2 問題設定

書き言葉→話し言葉変換を行うことなどを説明する。またこの実験における話し言葉、書き言葉の定義についても言及しておく。

7.3 実験) 書き言葉→話し言葉のスタイル変換

7.3.1 実験概要

データセット、モデルの説明を行う。

7.3.2 実験結果

実験結果を示す。

7.3.3 考察

考察を示す。

8 CoLA タスクを応用した対話システムのエラー検知

8.1 問題設定

深層学習を用いた対話モデルや、文生成のモデルを用いる際に出てしまう可能性のある不自然な文を検出するという問題設定について説明する。

8.2 実験) 対話システムのエラー検知

8.2.1 実験概要

BERT を用いて実験したことを示す。(このモデルを作成するにあたり文の自然さを評価するための CoLA タスクというものに注目し、これを解いている BERT と呼ばれるモデルを用いる。)

8.2.2 実験結果

実験結果を示す。

8.2.3 考察

考察を示す。

9 付録

この付録の存在意義について説明する。(論文の補足であることを説明する)

9.1 対話システムの関連研究

この章では 2 で引用した対話システムのうち、Sounding Board と Gunrock について詳細な説明を行う。りんなに関しては非公開情報が多いため説明を省略する。

9.1.1 Sounding Board

Sounding Board Fang et al. 2018

9.1.2 Gunrock

Gunrock Chen et al. 2018

9.2 日本語データの取り扱いについて

9.2.1 単語分割

単語分割

9.2.2 形態素解析

9.2.3 NER

9.2.4 Word Piece

Word Piece

9.2.5 Sentence Pieces

Sentence Pieces

9.2.6 Skipgram

Skipgram

9.2.7 CNN-RNN

CNN-RNN

9.3 質問文抽出を念頭においた不均衡分散・サイズの分類問題

9.3.1 画像データ

画像データ

9.3.2 文データ

文データ

9.4 機械翻訳システムを用いた対話

9.4.1 Seq2Seq Attention

Seq2Seq Attention

9.4.2 Transformer

Transformer

9.5 文のスタイル変換

9.5.1 Sequence to Better Sequence

Sequence to Better Sequence

9.5.2 CopyNet

CopyNet

9.5.3 Denoising Auto Encoder

Denoising Auto Encoder

9.6 CoLA タスクを応用した対話システムのエラー検知

9.6.1 BERT

BERT

10 結論

10.1 今後の課題

今回できなかった文生成の問題・論文に載せることのできなかった推論の内部状態の更新等について言及する。また精度向上や今後取り組みたい問題設定 (Unity など で 仮 想 世 界 を 作 り、そ の 中 で 対 話 を 行 え る よ う に す る エ ー ジェ ント 作 成 し た い 旨) に つ い て 話 す。

References

- Chen, Chun-Yen et al. (2018). *Gunrock: Building A Human-Like Social Bot By Leveraging Large Scale Real User Data*.
- Chu, Hang, Daiqing Li, and Sanja Fidler (2018). “A Face-to-Face Neural Conversation Model”. In: eprint: arXiv:1812.01525.
- Fang, Hao et al. (2018). *Sounding Board: A User-Centric and Content-Driven Social Chatbot*. eprint: arXiv:1804.10202.
- Serban, Iulian Vlad et al. (2016). *A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues*. eprint: arXiv:1605.06069.
- Sordoni, Alessandro et al. (2015). *A Hierarchical Recurrent Encoder-Decoder For Generative Context-Aware Query Suggestion*. eprint: arXiv:1507.02221.
- Wo, Xianchao et al. (Mar. 2016). りんな：女子高生人工知能. 言語処理学会 第 22 回年次大会 発表論文集. Microsoft Japan Inc.
- 逸子, 藤村, 大曾 美恵子, and 大島 デイヴィッド義和 (2011). 言語研究の技法：データの収集と分析. Ed. by 藤村逸子、滝沢直宏. ひつじ書房.

11 C++ コードを Emacs + Org mode で書く

```
1 std::cout << greet << std::endl;
```

Listing 1: テストコード

Hello World

```

1  (setq org-latex-listings 'minted
2      org-latex-packages-alist '((" " "minted")))
3
4  (require 'org)
5  (require 'ox-latex)
6  (require 'ox-bibtex)
7  (require 'org-ref)
8
9  (setq org-latex-toc-command
10     "\\pagenumbering{roman} \\tableofcontents
11     \\clearpage \\listoffigures \\clearpage")
12
13  (setq org-format-latex-options
14       (plist-put org-format-latex-options :scale 2.0))
15
16  (setq bibtex-completion-pdf-open-function 'org-open-file)
17
18  (setq org-latex-listings 'minted)
19
20  (setq org-latex-minted-options
21       '(("frame" "lines") ("linenos=t")
22         ("obeytabs") ("tabsize=4")))
23
24  (setq org-latex-pdf-process
25       '("latex --shell-escape --kanji=utf-8 %f"
26         "latex --shell-escape --kanji=utf-8 %f"
27         "biber %b"
28         "latex --shell-escape --kanji=utf-8 %f"
29         "dvi2pdf %b"))
30
31
32  (org-babel-do-load-languages
33   'org-babel-load-languages
34   '((C . t)))

```

Listing 2: init.el

```

1  #+CAPTION: テストコード
2  #+header: :includes <iostream>
3  #+header: :var greet="Hello World"
4  #+begin_src C++ :exports both :results latex :cache yes
5  std::cout << greet << std::endl;
6  #+end_src
7
8  #+RESULTS[72a19f4e33f7ff6180eab6e0536194c82f003d2c]:
9  #+BEGIN_EXPORT latex
10 Hello World
11 #+END_EXPORT

```

Listing 3: 書き方