クラウドソーシングを用いた対話コーパス構築

横野 光, 高橋哲朗
株式会社 富士通研究所
{yokono.hikaru,takahashi.tet}@jp.fujitsu.com

1 はじめに

言語処理の研究やシステム開発では，現象の分析やモデルの構築のために対象としているドメインのコーパスが必要となる．これらの目的のために現代日本語書き言葉均衡コーパス (BCCWJ) などのような様々なコーパスが作成，公開されている．また，近年では EC サイトのレビューや SNS のコンテンツなどの CGM を含む Web 上のデータが大量に存在し，これらを使用した研究も行われている．

しかし，既存のコーパスに存在しないようなドメインを対象とする場合は，コーパスを新たに構築する必要がある．特に既存のコーパスの多くは新聞記事などの書き言葉による文書であり，対話を対象とした場合では開発用のコーパスを作成することがより大きな問題となることが多い．

近年，クラウドソーシングを利用したデータ作成手法が注目されている．クラウドソーシングでは作業者の質を担保することが困難であるが，一般的なデータ作成に比べてコストが低いため，比較的容易なタスクで大量のデータが必要である場合に用いられることが多くなっている．しかし，多くのクラウドソーシングサービスでは基本的に作業者が単独で実施可能なタスクを想定しているため，対話のように複数人の作業者を必要とするタスクを行うにはそれに合わせた対応を取る必要がある．

そこで本研究では，この問題に対して対話コーパス作成タスクをクラウドソーシングに適した形に分割することで，作成コストを下げることを目指す．実際に提案手法によって対話コーパスを作成し，一般的な方法によって作られた対話コーパスとの比較を行う．

2 関連研究

対話コーパスの基本的な構築では作業者を集め管理するというコストが生じるため，データ収集においてその負担を減らしたり，対話に近いようなデータを使用するなど，様々な手法が提案されている．東中らは雑談対話 API を利用したシステムと作業者との対話からなる雑談対話コーパスを構築している [1]．叶内らはゲーミフィケーションを採用し，ユーザからの自発的な対話データ収集法を提案している [2]．塚原らは作業者の管理をコミュニケーションツールによって行いクラウドソーシングを用いた対話コーパスの作成を行っている [3]．

また，話し言葉に近い発言の大規模コーパスとして Twitter[1] の tweet や reply のデータが注目されており，例えば，杉山らは tweet から発話に対する応答文の生成を行っている [4]．

コーパス作成の手段をはじめとして，クラウドソーシングは近年注目されており，HCOMP-16[2] のようなワークショップも開催されている．例えば，Huang らはクラウドソーシングによってユーザの質問に回答するというシステムを構築している [5]．Paperno らはクラウドソーシングを利用して自然言語理解タスク用のデータセットを構築している [6]．クラウドソーシングでは作業品質の維持が重要な問題であり，それに向けての研究も行われている (cf. [7])．

3 コーパス作成方法

一般的な対話コーパスの構築手法は，複数の作業者を同時刻に作業空間に配置し，そこで実際に対話を行ってもらいデータを収集するというものである．ここで，作業空間とは物理的に同じ場所であったり，チャットシステムなどを利用する場合ではそのチャットシステムのことを指す．本稿ではこのように作業領域に同時刻に作業者を割り当てることを “**作業者間の同期を取る**” と呼ぶ．対話コーパス作成では基本的に作業者間の同期を取る必要があるため，テキストデータに対するタグ付けなどのような一般的なコーパス作成に比べて，作業者の管理のコストなどが余分にかかることになる．

近年，クラウドソーシングによるデータ作成が注目されているが，一般的なクラウドソーシングサービスでは各作業者に独立したタスクを割り当

[1] https://twitter.com
[2] http://www.aaai.org/Library/HCOMP/hcomp16contents.php

てるため，対話コーパス作成にそのまま利用することは困難である．

そこで本研究ではこの問題に対して，対話データの作成を “与えられた背景，文脈に対して，それに続く発話を作成する” という個別のタスクに分割することによる対話コーパス構築手法を提案する．

2 人の参加者による対話のデータ作成の場合，通常の対話データ作成では 2 人の作業者を対話の参加者として 1 対話に割り当てる．これに対して，提案手法では 1 人の対話の参加者の発話を複数人の作業者が担当する．つまり，対話の参加者を u_1，u_2 とすると，作業者は u_1 か u_2 のいずれかの立場で発話を生成することになる．ここで u_1，u_2 それぞれの背景を $BG(u_1)$，$BG(u_2)$ とする．

対話は u_1 から発話するとし，次に u_2 の発話，と交互に発話するという設定にする．対話の最初の作業者は $BG(u_1)$ を参照して u_1 の発話 U_1 を生成する．次に割り当てられた作業者は u_2 の立場で $BG(u_2)$ とこれまでの発話 ($\{U_1\}$) を参照して発話 U_2 を生成する．この過程を繰り返すことで u_1 と u_2 の対話 $\{U_1, U_2, \dots\}$ を疑似的に生成する．

4 評価

提案手法では，2 人による対話において本来それぞれ 1 人の対話参加者によって生成される発話を複数人の作業者によって作成する．これによってデータ作成のコストを削減することが可能になると考えられるが，一方で，1 人の発話を複数人で作成しているため，発話の一貫性が下がるといった通常の対話とは異なる性質を持つ可能性が出てくると考えられる．

そこで実際に提案手法で作成したコーパスが一般的な手法によって作成された対話コーパスとどの程度異なるかについての分析を行った．

4.1 コーパス構築

事例として “不動産屋に物件を探しに来た客と不動産屋との対話” という内容の対話データを作成した．客側に割り当てられた作業者には図 1 のような物件を探しに来た背景が与えられ，これに基づいて求めている物件に関する発言をする．不動産屋側に割り当てられた作業者にはこの背景は与えられず，客側の発言から要望にあった物件を検索するための情報を引き出すことが求められる．

このタスク設定において，多様なデータを作成するためには様々な種類の背景を用意することが

図 1: 客側に与えられる背景の例

> 2 年ほど付き合った彼氏と同棲することになり、これまでのワンルームから引っ越すことになった女性。これを機に料理に力を入れたいため、コンロが多く使いやすいキッチンがある物件を希望している。

重要となる．そこで対話データ作成に先立ち，不動産屋に訪れる人の背景をクラウドソーシングによって作成し，そこから著者らが実際にデータ作成に使用する背景を選択した．

データ作成では不動産屋側の作業者が実際に客側の要望から物件を検索することは行わず，不動産屋側が物件検索に必要な情報を客側から引き出せたと判断した時点で対話終了とする．また，客側，不動産屋側のいずれかがこれ以上対話を続けられないと判断した場合は “不適切” と，客側，不動産屋側あわせて 20 発話に達した場合は “中断” とタグ付けして対話終了とするように指示した．

提案手法では対話における各発話を個々の作業者が作成するため，1 つの発話に対して複数の作業者に割り当てることもできる．これによって対話のある時点までを共有し，そこから分岐が起きたという状況を再現することができる．原理的には全ての発話に対して複数の作業者を割り当てることができるが，発散してしまうため，今回は図 2 に示すように途中までは 1 発話に対して 3 人の作業者を割り当て，そこから先は 1 発話に 1 人の作業者を割り当てた．

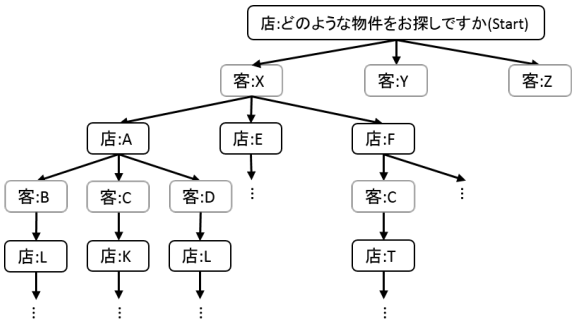

図 2: 1 背景における作業者割り当ての例 (枠内の英字は作業者の ID を表す)

作業者の割り当てに関しては以下の条件を満たすようにした．

- 同一背景内では作業者は不動産屋側か客側のいずれかしか担当できない
- 1 つの対話において作業者は同じ作業者の後に発言できない

後者の条件は 1 つの対話において作業者ペアが固定されるのを防止するためであるが，これは実験的に複数の作業者が同じ側を担当するという状況を作成するために設定したものであり，実際の運用時には作業者を割り当てるコストが変わらないのであれば不要である．

疑似対話がどの程度実際の対話と似ているかを評価するため，比較対象として同様の課題で客側と不動産屋側の作業者をそれぞれ 1 人に固定した場合の対話データを作成した．このデータに関してもクラウドソーシングによって作成している．以降，提案手法によって作成した対話データを**疑似対話**，比較用に作成した対話データを**チャット対話**と呼ぶ．

4.2 チャット対話との比較

作成した疑似対話コーパスの例を図 3 に，統計を表 1 に示す．

不動産屋:どのような物件をお探しですか？
客:しっかりと料理をしたいので、キッチンが広くて、コンロの数が多い物件が良いです。
不動産屋:コンロは何口必要ですか？
客:コンロは 3 口はほしいです。
不動産屋:IH クッキングヒーターやガスコンロ等、コンロにご希望はございますか？
客:光熱費の節約になって安全なコンロがよいのですが、ガスと IH ならどちらがよいですか？
不動産屋:しっかり料理をしたいのでしたらガス、安全性を考えるならＩＨが思うのですが、小さなお子さんはいらっしゃいますか？
客:子供はおりませんが、実家のキッチンが IH クッキングヒーターで非常に使いやすいので、可能であれば IH クッキングヒーターが 2 口、通常の電熱器タイプが 1 口ついたキッチンを希望します。
……

図 3: 疑似対話の例 (一部)

不適切と判断された対話数が 3 割弱存在しているが，後述するように著者らがデータを見て実際に不適切と判断した対話の数は少なかった．これは作業者にとっては発話を考えるよりは不適切を選択する方が容易であるため，そちらを選んだ作業者がいたからだと考えられる．これに対しては，指示において不適切と判断する場合にはその理由を記述するなど，発話の作成と同じ程度の負荷とすることで解消できると考えられる．

表 1: 疑似対話の作成統計

項目	値
総対話数	270
途中で不適切と判断された対話数	88
発話制限に達して中断された対話数	33
1 作業者当たりの発話数	1.012

疑似対話とチャット対話との比較を表 2 に示す．

表 2: 疑似対話とチャット対話の比較

	疑似対話	チャット対話
平均発話数	11.73	19.30
1 発話当たりの平均文字数	27.57	14.99
1 対話当たりの平均語彙数	88.10	75.85
1 対話当たりの共通語彙数	22.72	17.11
共通語彙の割合	0.246	0.223

チャット対話に比べて疑似対話の方が 1 対話の発話数が多く，1 発話当たりの文字数も多くなっている．これは，実際の対話と異なり疑似対話では作業者は基本的には次の発話を担当することがないため，次にどのようなことを発言すればいいか，といった対話の戦略を立てることが困難となり，与えられた時点で発話できることをまとめて記述しているからだと考えられる．また，タスクが “これまでの文脈を読んで，次の発話を考える” という作文課題になっているというのも原因と考えられる．

不動産屋側の発話と客側の発話において，どの程度の語彙が共通して出現しているかに関しては，比較的同じような割合となっている．

疑似対話とチャット対話のそれぞれ 90 件に対して，対話中の発話が不適切になっていると思われる箇所とトピックの境界をアノテーションした．不適切さのアノテーションに関しては，局所的な文脈での破綻，大域的な文脈での破綻の両方を対象とし，誤字や小さな誤りなどは元の文が推測できる場合，不適切とはみなしていない．トピックの境界に関しては，不動産の物件検索で良く用いられるカテゴリである “間取り” や “家賃” などをトピックの単位とした．アノテーションは 1 人の作業者で行っている．

アノテーション結果を表 3 に示す．

対話当たりのトピック数は，表 2 に示しているように対話当たりの発話数に差があるため，これに伴い差が生じているが，1 トピックあたりの発話

表 3: アノテーション結果

	疑似対話	チャット対話
1 対話当たりの平均トピック数	4.644	7.011
1 トピック当たりの平均発話数	2.808	2.740
不適切と判定した発話を含む対話数	14	8

数は差がなく，トピック内でのやり取りとしては実際の対話と大きな異なりはないと考えられる．

不自然さに関しても，複数人で 1 人の対話参加者の発話を作成しているにもかかわらず，数は多くなかった．しかし，アノテーションした 90 件のうち 50 件に対して作業者を追加してアノテーションし，片方を正解とみなしたときの一致率は約 0.5 であった．このアノテーションを行う前に，基準を合わせるために別のデータに対して作業者 2 人で協議しながらアノテーションを行ったが，一致率は低かったため，より詳細な基準を用意する必要がある．

5 おわりに

本稿では，クラウドソーシングサービスを利用した対話コーパスの作成に対して，対話参加者の背景とこれまでの対話履歴から次の発話を生成するというタスクに分割するという手法を提案した．

疑似的に対話を作成しているため，自然な状況において作成された対話データに比べて本手法によって作成した対話データは，前の発言の取り消しが行われない，同じ対話参加者が連続して発話しない，など現れる現象に制約があるが，1 人の発話を複数人で担当したにも関わらず，不自然な箇所が少ないものとなった．

しかし，どのような対話も提案手法で作成できるというわけではない．提案手法では対話参加者の背景や対話に必要な知識が作業者間で共有できる必要がある．今回作成したデータの設定では，客がどのような背景を持っているかというのが客側を担当した作業者で共有されており，また，不動産屋でのやり取りという必要な知識が限定しやすい状況であったため，実際の対話に近いデータが作成できたと考えられる．雑談のように対話にどのような知識が必要かがあらかじめ想定できないようなタスクでは提案手法を適用することは困難であると考えられる．

提案手法はコーパス作成コストを削減できるというメリットがあるが，それだけではなく，対話履歴に対する発言の生成，というタスクにしているため，同じ対話履歴に対して複数の作業者を割り当てることができる．これによって，任意の時点までは共通であるが，それ以降が異なるといった対話を容易に作成することができる．このようなデータを作成することで，対話の流れにどの程度多様性があるかを分析することが可能であると考えられる．

本稿では小規模のコーパスを作成して評価を行ったが，今後は提案手法によって大規模なコーパスを作成し，データの評価を行うとともに，より自然な対話を作成可能なタスクの設計を行う予定である．

参考文献

[1] 東中竜一郎，船越孝太郎: Project Next NLP 対話タスクにおける雑談対話データの収集と対話破綻アノテーション，人工知能学会言語・音声理解と対話処理研究会 第 72 回研究会，pp. 45–50 (2014).

[2] 叶内　晨，小町　守: ゲーミフィケーションを利用した効率的な対話ログ収集の試み，電子情報通信学会信学技報 NLC2016-30 (2016).

[3] 塚原裕史，内海　慶: オープンプラットフォームとクラウドソーシングを活用した対話コーパス構築方法，言語処理学会第 21 回年次大会，pp. 147–150 (2015).

[4] 杉山弘晃，目黒豊美，東中竜一郎，南　泰浩: 任意の話題を持つユーザ発話に対する係り受けと用例を利用した応答文の生成，人工知能学会論文誌，Vol. 30, No. 1, pp. 183–194 (2015).

[5] Huang, T.-H. K., Lasecki, W. S., Azaria, A. and Bigham, J. P.: “Is There Anything Else I Can Help You With?” Challenges in Deploying an On-Demand Crowd-Powered Conversational Agent, *Proceedings of The Fourth AAAI Conference on Human Computation and Crowdsourcing*, pp. 79–88 (2016).

[6] Paperno, D., Kruszewsk, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G. and Fernández, R.: The LAMBADA dataset: Word prediction requiring a broad discourse context, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1525–1534 (2016).

[7] 梶村俊介，馬場雪乃，梶野　洸，鹿島久嗣: 列挙型クラウドソーシングタスクのための品質管理法，人工知能学会論文誌，Vol. 31, No. 2 (2016).