

修士論文 2011年度(平成23年度)

会話コンテキスト抽出のための
スマートフォンを利用したユーザグルーピング

慶應義塾大学 政策・メディア研究科
唐津 豊

karasu@sfc.keio.ac.jp

修士論文要旨 2011 年度 (平成 23 年度)

会話コンテキスト抽出の為の スマートフォンを利用したユーザグルーピング

センサデバイスの高機能化によって、実世界上の出来事や人々の周囲の状況に関するコンテキストをセンサ情報に基づいて解析する技術の研究開発が盛んに行われている。コンテキストの中でも、人々が Face-to-Face で行う会話に関するコンテキストはコンテキストアウェアシステムの分野において様々なサービスの実現を促進させることが期待される。

Face-to-Face の会話において、人々が生活する過程で自然に発生する懇談や相談といった会話は人々の日常生活内における会話の大部分を占める。このような会話についてコンテキスト抽出を行う場合、人々の日常生活内で突発的に発生する会話を検知し、その会話の参加者を特定する必要がある。これに関して、大きく 2 つの課題を解決しながらコンテキスト解析を行う必要がある。

まず、これらの会話は発生場所や発生時間を事前に予測することができないため、環境に敷設されたセンサネットワークのような情報収集設備による基礎データの収集は効果が期待できない。そのため、ユーザがセンシング端末を携帯し、周囲の情報に関して常時センシングを行う必要がある。また、日常生活内で自然発生する会話は、会話を行なっている人々のすぐ近傍に会話への非参加者が存在する場合が度々発生する。この近傍の非参加者を会話の参加者と誤検知しないように会話参加者を特定しなくてはならない。そのため、位置情報や人々の相互近接情報といった空間的な会話参加者検知は効果が期待できない。

本論文では自然発生する会話のコンテキスト解析のため、“誰が誰の発言に何度応答したか”という会話応答頻度に基づいて会話参加者の特定を行う機構 Milvus を提案する。Milvus はこれらの会話が参加者間で双方向的に発話と応答を繰り返すことに着目し、頻繁に会話応答を行う人々を互いに同じ会話へ参加していると判断して、会話毎に参加者のグルーピングを行う。これにより、“いつ、どこで、誰と会話を行ったか”という会話コンテキストの抽出が可能となる。Milvus はセンシング端末としてスマートフォンを想定することで、ユーザが日常生活には本来余分なセンシング端末を携帯する負担の軽減を図る。また、会話応答頻度に基づいた抽出処理を行うことで、会話参加者に近傍する非参加者を区別することに効果が期待できる。

本研究において、Milvus の会話参加者グルーピングの精度評価実験を実施したところ、会話グループが同時並行的に発生するセミパブリックな空間内で想定するアプリケーションでの利用に十分な精度でグルーピングを行う事ができた。さらに、本実験環境以外において会話コンテキストの抽出を行うためには、システムの利用可能環境の拡大と精度向上は不可欠である。今後の改良案としては音声情報と共に複数のセンサ情報を利用した精度補完、ソーシャルネットワークサービスから得られる友人関係情報といった定性的な情報に基づくグルーピング処理の柔軟性向上といった改良案が挙げられる。

キーワード：

コンテキストアウェアシステム, 会話グルーピング, アドホック会話, スマートフォン

慶應義塾大学 政策・メディア研究科

唐津 豊

Abstract of Master's Thesis Academic Year 2011

User Grouping Scheme for Ad-hoc Conversations with Smartphones

Researches and developments to analyze contexts that are information of events on the real world and situations surrounding persons are one of the most important issues in ubiquitous computing. The one of the useful for realizing novel applications of these is face-to-face conversation contexts.

In face-to-face conversations, ad-hoc conversations, for example counseling, discussions, and chatting occurring naturally, are made up of a majority of daily conversations. When the context analysis system deals contexts of these conversations, it is necessary for the system to detect unexpectedly occurred conversations and to identify participants in each conversation.

The ad-hoc conversation context analysis has two problems as follows. Firstly, the system needs to constantly acquire information surrounding each user because sensor networks fixed in living spaces are inadequate to the conversations unpredictable when and where occur. Secondly, it is necessary for the systems to distinguish non-participants users who stay in close to conversation groups between participants.

We propose “Milvus” that a mechanism to analyze the ad-hoc conversation context based on Conversation Response Frequency (CRF) that is information how many times does some one response to the remarks of whom. The conversation participants alternately speak in the ad-hoc conversation group. Milvus detects users often speaking alternately from CFR as the conversation group. Milvus leverages smartphones as mobile sensing devices in order to acquire sensor data surrounding each user constantly. In addition, CRF allows distinguishing between conversation participants and non-participants nearby the group.

Experiments show Milvus achieves the grouping accuracy of 70% in assumed environments like semipublic office. In future work, we improve proposing grouping scheme to complement grouping results leveraging multiple sensor data and qualitative data gathering from social network web services.

Keywords

Context-aware System, Conversation Grouping, Ad-hoc Conversation, Smartphone

Yutaka Karatsu

**Graduate School of Media and Governance
Keio University**

目次

第 1 章	序論	1
1.1	背景と目的	1
1.2	アプローチ	2
1.3	本論文の構成	3
第 2 章	会話コンテキスト情報の抽出	4
2.1	コンテキストウェアシステム	4
2.2	シナリオ	5
2.3	アドホック型会話	6
2.3.1	日常生活内における会話分類調査	7
2.3.2	本論文の対象領域	9
2.4	目的と課題の整理	9
2.4.1	センサ情報常時取得問題	9
2.4.2	同一空間内非参加者問題	10
2.5	機能要件	10
2.5.1	自然発生する会話の検知	10
2.5.2	非参加者の排除	11
2.6	本章のまとめ	11
第 3 章	会話応答頻度情報によるユーザグルーピング	12
3.1	Milvus	12
3.1.1	アプローチの詳細	12
3.1.2	手法のデザイン	13
3.1.3	想定環境	13
3.1.4	処理の流れ	14
3.2	発話区間の取得	15
3.3	会話応答頻度の算出	16
3.3.1	会話応答頻度	16
3.3.2	対象ユーザの絞り込み	17
3.3.3	会話応答頻度抽出アルゴリズム	25
3.4	会話応答頻度に基づくユーザグルーピング	25
3.5	関連研究	28

3.5.1	空間的基準によるグルーピング	28
3.5.2	グルーピング専用端末を要する手法	29
3.5.3	長期収集情報に基づく行動傾向解析	30
3.5.4	音声解析による手法	30
3.6	本章のまとめ	31
第4章	設計・実装	33
4.1	設計	33
4.1.1	ハードウェア構成	33
4.1.2	ソフトウェア構成	33
4.1.3	クライアントソフトウェア	34
4.1.4	センタユーザ・周辺ユーザ定義生成部	35
4.1.5	会話応答頻度算出部	35
4.1.6	会話グループ判定部	35
4.2	実装	36
4.2.1	実装環境	36
4.2.2	クライアントソフトウェア	37
4.2.3	Milvus	40
4.3	本章のまとめ	41
第5章	会話コンテキスト情報の利用	42
5.1	会話コンテキストの利用	42
5.1.1	サービス対象の指定	42
5.1.2	仮想情報の注釈	43
5.1.3	日常生活内のソーシャルネットワーク情報の蓄積	43
5.2	会話コンテキストに基づくコンテンツ共有システムの試作	44
5.2.1	試作アプリケーションの動作	44
5.2.2	アプリケーションの実装	44
5.3	本章のまとめ	47
第6章	評価	48
6.1	Ground Truth に基づくグルーピング精度評価	48
6.1.1	手順	48
6.1.2	実験環境と装置	49
6.1.3	結果	49
6.1.4	考察	50
6.2	試作アプリケーションによるユーザビリティ評価	55
6.2.1	実験計画	55
6.2.2	実験環境と装置	56
6.2.3	手順	56
6.2.4	被験者	56

6.2.5	結果	56
6.2.6	考察	58
6.3	本章のまとめ	58
第7章	結論	59
7.1	今後の展望	59
7.1.1	多様なセンサ情報の利用による精度向上	59
7.1.2	定性的情報に基づくグルーピング処理の補完	60
7.1.3	会話グループ情報取得インタフェースの充実	60
7.2	まとめ	61
	参考文献	64
付録 A	ユーザスタディ評価用紙	66
付録 B	ユーザビリティ評価アンケート	70

目次

2.1	本論文が定義する会話形体の模式図	7
2.2	調査結果:会話参加者数と会話突発性	8
2.3	調査結果:会話時間と会話突発性	8
3.1	セミパブリックな空間の例	14
3.2	発話者と応答者の時系列的な遷移	17
3.3	会話応答者の誤検知	18
3.4	高頻度発話者	19
3.5	発話区間のオーバーラップと同時発話者の選出	21
3.6	発話オーバーラップと所属グループ	22
3.7	オーバーラップ最多者と次点者間のオーバーラップ差分が小さい場合の3ユーザの所属グループ	22
3.8	Sociometer 装着図 (引用:[1])	29
3.9	DarwinPhones フレームワークの基本概念 (引用:[2])	32
4.1	ハードウェア構成図	34
4.2	会話グルーピング処理に関するソフトウェア構成図	35
4.3	使用端末	36
4.4	音声区間検出関数	38
4.5	発話区間情報の生成	38
4.6	センサ情報更新クエリ文字列	39
4.7	グループ情報取得クエリ	39
4.8	グルーピング結果	40
5.1	アプリケーション構成図	45
5.2	試作アプリケーションスクリーンショット	46
6.1	S103 号室での実験の様子	49
6.2	Ground Truth に基づくグルーピング精度実験結果	50
6.3	実験 E1 実験結果内訳抜粋:実験時間 0:00 から 10 分間の Ground Truth 情報とグルーピング結果の比較	53
A.1	アンケート	66
A.2	実験配布資料 (1/2)	67

A.3	実験配布資料 (2/2)	68
A.4	被験者情報シート	69
B.1	ユーザビリティ評価アンケート	71

表目次

2.1	アンケート質問項目	8
4.1	センシング端末実装環境	37
4.2	サーバシステム実装環境	37
5.1	試作アプリケーション：クライアント実装環境	46
5.2	試作アプリケーション：サーバ実装環境	46
6.1	操作時間の比較結果	57
6.2	送信対象指定時負荷の比較結果	57
6.3	会話コンテキスト情報の有効性	58

第 1 章

序論

1.1 背景と目的

近年の携帯端末やセンサデバイスの進歩によって、実世界上の出来事や人々の周囲の状況に関する情報であるコンテキスト情報の解析技術の研究開発が盛んである。これらの技術を用いてコンテキスト情報を抽出、蓄積、配信することによって、特定の出来事の発生に合わせてアプリケーションを起動する、人々の周囲の状況にそぐわしいサービスを自律的に提供するといった、実世界指向なコンテキストウェアシステムを実現できる。このコンテキスト情報の中でも“いつ、どこで、誰と会話を行ったか”という、人々が日常生活内で行う Face-to-Face での会話に関する情報は様々なアプリケーションへの応用が期待される。会話に関するコンテキスト情報をネットワーク上のストレージに蓄積することで、煩わしい操作を介さずに会話相手にデジタル情報を共有できる情報共有アプリケーション、会話履歴情報をクエリとした情報検索サービス、実世界上のソーシャルネットワーク解析といったサービスを実現できる。本論文では、これらのようなアプリケーションが必要とする“いつ、どこで、誰と会話を行ったか”というコンテキストを**会話コンテキスト**と呼ぶ。

会話コンテキスト情報を抽出するためには、人々の日常生活内で突発的に発生する会話を検知し、その会話の参加者を特定する必要がある。1つの会話に多くの人々が参加する講演や授業、事前に会話の発生を予測できる定例会と異なり、日常生活内で発生する懇談や相談といった会話は発生場所や発生時間を事前に予測することができない。そのため環境側に敷設されたセンサネットワーク、デバイスインフラから得られる情報やユーザのパーソナル情報に基づいて会話を検知することは困難であり、人々の周囲の状況に関する情報をウェアラブルなセンサデバイスを使用して常に取得し続ける必要がある。本論文ではこのような事前予測が難しい、少人数の人々の間で突発的に発生する会話を**アドホック型会話**と呼ぶ。

一方で、アドホック型会話の会話コンテキスト抽出は、会話の近傍に存在する非参加者を会話グループから適切に排除できる必要がある。日常生活空間内の会話は発生場所の近傍に会話への非参加者が存在することが頻繁に起こりうるからである。同一空間内で複数の会話グループが発生する場合など、アドホック型会話では会話に参加していない人が存在する場合が多い。そのため従来の集団検知技術に頻繁に用いられる、GPS 位置情報やウェアラブルデバイスの短距離無線電波強度に基づくデバイス間距離情報からこのような非参加者を区別することは困難であり、会話に無関係な人を参加者として判定してしまうことで的確な会話コンテキスト抽出を妨げる恐れがある。

本研究では、日常生活内で発生するアドホック型会話の会話コンテキストを抽出することを目的とする。また、アドホック型会話の会話コンテキストをコンテキストウェアアプリケーションが利用可能にする。

1.2 アプローチ

そこで本研究は、会話参加者の発言に対する会話相手の受け答え頻度に基づいてアドホック型会話の参加者を特定し、その結果を会話コンテキストとして抽出する機構“Milvus”を提案する。アドホック型会話は会話内の話者と聴者が頻繁に入れ替わりながら、会話参加者の発言に対して他の参加者が応答することを繰り返すことで進行する。そのため、ある話者の発言に対して頻繁に応答する人は話者と同じ会話に参加していると考えられる。Milvusはこの仮説に基づき、ある会話参加者が単位時間内で行った発言に対して頻繁に応答した人間をその話者と同じ会話に参加していると認識して会話参加者の特定を行う。本論文では、アドホック型会話の会話コンテキスト抽出を実現するために会話毎に参加者を特定することを**グルーピング**と呼び、グルーピングに利用する会話参加者の発言に対する受け答え頻度を**会話応答頻度**と呼ぶ。

Milvusでは携帯端末のマイクで取得した音声解析して会話応答頻度を取得する。音声データから端末所有者が単位時間中に発話し続けた時系列区間を取得し、この情報を近傍にいるユーザ同士で比較することで会話応答頻度をユーザ毎に算出する。以下にMilvusの設計指針を挙げる。

- 日常的な携帯端末の利用
アドホック型会話は会話の発生場所及び時間を予測することが困難なため、ユーザ周囲の情報を常にセンシングする必要がある。そのためユーザはセンシングデバイスを身に付けなければならない。この時、グルーピングに最適なセンサ情報を得るため、グルーピングに特化した専用デバイスを身に付けさせることも可能である。しかし、専用デバイスの利用はユーザが日常生活内で携帯するデバイスに本来必要ない余分なデバイスを追加することとなり、ユーザに負担を強いることとなる。そこで、本研究では近年急速に普及しているスマートフォンで取得できる情報に基づいてグルーピングを行うこととする。
- アドホック型会話の検知
アドホック型会話は少人数の会話参加者で構成されるため、アーバンセンシングで用いられる人々の位置情報遷移といった情報からの発生検知は困難である。また、発生場所、発生時刻を限定できないため、スケジュールアプリやTodoリストアプリ等に登録されるパーソナルデータや、オフィス内の鍵システムから得られる入室記録のような環境敷設型情報収集機構から得られる情報による検知も難しい。本研究では、スマートフォンのマイクから得られる音声データに基づいて継続的に発話区間情報を取得する。さらにこの情報を短期定期的にコンテキスト解析サーバへ送信し、サーバ上で解析処理を短周期で実行する。これによって、リアルタイム性を維持しながらアドホック型会話の検知を行う。
- 近傍非参加者の排除
アドホック型会話のグループ周囲には会話に参加していない無関係な人々が存在する場合が多い。会話コンテキストを抽出する上で、グループ近傍の会話非参加者をグループの参加者であると判断する誤検知を防ぐ必要がある。既存研究ではユーザの絶対位置情報、短距離無線電波強度から得られるユーザ間近接値、携帯端末で取得した音声データの相互相関情報といった情報をグルーピングに利用する手法が提案されているが、これらの手法は総じてユーザ同士の近接度合いを基準としてグルーピングを行なっている。そのため、グループ近傍の非参加者の誤検知が高頻度で起こりうる。本研究ではユーザの発話区間情報に基づき会話応答頻度情報を算出しグルーピングの指標として用いることで、非参加者の誤検知を低減する。

会話応答頻度の取得及び会話コンテキスト抽出処理は、会話応答頻度情報取得のために各ユーザの発話区間情報を比較する必要があること、コンテキスト算出処理のリアルタイム性を考慮する必要があること等の理由から、端末で取得した情報を遠隔のコンテキストサーバに送信し、遠隔サーバ上で解析処理を行うサーバクライアント型のアーキテクチャを採用した。また、会話応答頻度取得のための話者認識処理はシステムユーザのプライバシーに配慮するために、高負荷な解析処理を目的として音声データを外部サーバにストリーミングするような音声の外部送信は行わない。携帯端末内で音声から特徴データに加工した上で、特徴データのみをコンテキスト解析サーバに送信する。さらに同様の理由で、音声自然言語解析による話者認識及びグルーピング手法も対象としない。

1.3 本論文の構成

以下に本論文の構成を示す。2章で本研究が対象とする領域、アプリケーションシナリオをユーザスタディ実験で得られた知見で補足しながら説明し、本研究が満たすべき機能要件について言及する。次に3章でMilvusの会話応答頻度情報の取得アルゴリズム、及びアドホック型会話におけるグルーピング手法について説明する。さらに関連する既存手法及び研究について言及した上で、本論文の提案システムと比較する。4章でMilvusを組み込んだ会話グルーピングシステムの設計及び実装について言及する。さらに、5章ではMilvusで解析された会話コンテキストに基づくコンテキストアウェアシステムの例として試作した、会話グループ内におけるコンテンツ共有システムについて説明する。6章で本研究が提案するグルーピングアプローチの定量的評価と共に、グルーピングシステムが抽出した会話コンテキストを使用した試作アプリケーションを利用してコンテキスト抽出機構の有用性についての定性的評価について実験と評価結果について説明する。最後に7章で今後の課題、展望を示した上で、本論文をまとめる。

第2章

会話コンテキスト情報の抽出

対人コミュニケーションにおいて、対面で行われる会話は日常生活内のコミュニケーションの大部分を占める。この会話の中でも突発的に自然発生するマイクロな会話について、会話コンテキストの抽出を行う。本章では本論文が想定しているコンテキストウェアシステムと利用シナリオについて説明した上で、それに対する課題と要件を明確にする。

2.1 コンテキストウェアシステム

多様なセンサの開発により、日常生活圏内にセンサ機器が浸透している。身近な家電製品やパーソナルコンピュータ（PC）には機器状態取得を目的としたセンサが内蔵されており、機器内部の温度や機器に与えられる振動を取得することで自律的に内部部品の稼働状態を制御することが可能になった。また、RFID タグや IC チップの他、短距離無線で取得した情報をベースステーションに定期的に送信する無線センサが電子機器以外の生活用品に取り付けられ、設置対象の状態をモニタリングすると共に人々にサービスを提供する製品が実用化されている。さらに近年では様々なセンサを内蔵しつつ、高速な演算能力とネットワーク伝送を兼ね備えるスマートフォンが急速に普及したことにより、人々が常にセンサを身に付けている社会が実現された。

このように人々の生活環境内に敷設された、または人々が常に携帯するデバイスに内蔵されたセンサで取得した情報に基づき、実世界上で起こった出来事や状況に関する情報の取得技術の開発が盛んに行われている。GPS から得られる位置情報や加速度センサから得られる被衝撃情報、温度や湿度など、各種センサで取得した情報を加工せずに得られるプリミティブな実世界情報は容易に取得することが可能である。さらに、これらの情報を複数組み合わせることで“誰が誰と同じ部屋にいるか”といった、より高次な実世界状況を取得することができる。実世界で起こる現象や人々の行動特性の傾向を調査し、これらのセンサ情報と関連付けてマイニングすることで、さらに抽象的な現象を扱うことも期待される。例えば、“街中でポップミュージックに興味のある人々がどこに集まっているか”といった複数の人間の活動に関する情報や“部屋の中がどのような雰囲気にあるか”といった心理的な状態など、実世界の状況に関する抽象的な情報の取得が可能となる。

これらの情報に基づいてサービスを提供することで、アプリケーションは人々に提供するサービスを取捨選択したり、自律的に動作して実世界上の状況に則したサービスの提供を実現できる。アプリケーションユーザの位置情報に基づいて現在位置に関連性の高い情報をレコメンドするシステムや、部屋の中の温度や湿度を基に自律的に稼働する電化製品等、プリミティブな実世界情報に基づくサービスの提供は日常生活でも多く見受けられる。実世界上の雰囲気や人々が形成している集団に関する情報など実世界の出来事や状況に基づいてサービス提供を行うことで、現在浸透しているアプリケーションはよりユーザセントリックに動作することが

可能になると期待される。

2.2 シナリオ

コンテキストウェアシステムに対して利用価値の高いコンテキスト情報は様々である。本論文ではその中でも人々が Face-to-Face で行うオーラルコミュニケーションに関するコンテキストに着目した。人々は会話を通じて実世界上で様々な情報を交換し、会話をしながら他者と協力して共通の課題の解決にあたるといったことを日常生活内において常に行なっている。さらに、個人の各会話行動を時系列的に調査すると、ある個人と他者との社会的な関係性を仮想空間内で扱うことが可能となる。このように日常生活内の会話は人々の相互インタラクションにおいて重要なファクターと言える。このことから会話に関するコンテキスト情報の解析、取得は様々なコンテキストウェアシステムを実現する上で重要な課題である。

本節では、本論文が想定する会話コンテキストの具体的な利用シナリオについて説明し、本論文が対象とする会話の体系を導く。

- サービス提供対象の指定

太郎さんは日頃から大学での研究活動に関連する Web 上の文献や Tips を掲載したウェブログの情報をブックマークとして収集し、自分のスマートフォンで閲覧できるようにしている。太郎さんが所属する研究会では授業時間外でも所属学生が研究室に集まって会話を楽しんでおり、研究会の同輩、後輩と会話が研究の話題に発展することも頻繁に起こる。太郎さんは研究について会話する時に、収集したブックマークの中から会話中の研究に関する参考資料を選んで会話相手に共有する場面に度々見受けられた。当初はその場で共有するコンテンツを電子メールに添付して会話相手に送信したり、Bump[3]のような物理的なインタラクションを利用した端末同士ペアリングによる情報共有アプリケーションを利用していたが、その都度端末を操作して送信先を指定しなければならず煩わしさを感じていた。さらに AirDrop[4]のような、自分の近傍にいる人々の端末に対してコンテンツをブロードキャストするアプリケーションを使用しても、会話に参加していない学生も送信対象に含まれてしまい、システムに対する煩わしさは拭えなかった。

そこで太郎さんはユーザの会話コンテキストに基づいてコンテンツの送信先を決定するアプリケーションを利用した。このアプリケーションは会話コンテキストからユーザが現在会話中の相手を判別してコンテンツの送信先を決定する。これにより太郎さんはコンテンツ共有の度に目の前にいる会話相手をアプリケーションから明示的に指定する必要がなくなり、煩わしさを感じる事がなくなった。さらにこのアプリケーションは Siri[5]、Google mobile app[6] としても実用化されている音声認識と会話コンテキストを組み合わせることで“皆にコンテンツを送信”とか“花子さんだけに送信”と、曖昧な音声指示でもコンテンツの送信を行うことができる。これは会話コンテキストによって事前にコンテンツの送信対象をユーザが関心を向けている会話相手のみに限定可能であるため実現された機能である。この会話コンテキストに基づいた共有アプリケーションを利用することにより、太郎さんは煩わしさを覚えずに情報共有を行うことが出来、会話をより楽しむことができた。

- 仮想空間内コンテンツの識別

太郎さんと同じ研究室に所属する花子さんは、太郎さんと研究室で会話している際に、自分の研究の参考になる論文のドキュメントファイルを送ってもらった。その場では軽く目を通しながら太郎さんと研

究について議論し、後日あらためて論文を精読することにした。数日後に花子さんは太郎さんから送ってもらった論文を自分の携帯端末から PC に転送しようとする。しかし、目的の資料は端末内の他のファイルに埋もれてしまい、どこに保存したか不確かであった。花子さんはファイルのタイトルを忘れてしまっていたので、更新日時を手がかりに太郎さんと話していた時間帯に保存されたファイルを順に閲覧して探していかなければならなかった。

そこで花子さんは自分の会話コンテキスト履歴を会話相手から共有されたコンテンツのアノテーションとして利用するファイル管理システムを導入した。このシステムは会話相手から情報を受信した際に、その時の会話コンテキストの ID を関連付けることで、会話相手の名前や会話をした場所をキーとしてファイル検索を実現するシステムである。このシステムの導入により、会話を行ったというイベントはファイル名やタイムスタンプよりも想起しやすく、研究会の友人から共有されたコンテンツの検索を今までより迅速に行うことができるようになった。

2.3 アドホック型会話

これらのシナリオで行われている会話は知人同士の懇談や相談といった日常生活内で自然発生する会話である。しかし日常生活内で起こる会話にはこれらの他に、大学で開講される授業やカンファレンスでの会議などのような、一度に大人数が 1 箇所に集まって特定の話者が大勢の聴者に話し続ける形体の会話や、研究会の定期ミーティングや企業内のチームが行う会議のように、会話の参加者間で事前に場所や時間を指定した上で会話を行う形体のものも存在する。

このように多様な形体を持つ対人コミュニケーションに対しての構造化に関する取り組みはこれまで幅広く行われている [7] [8] [9]。特に柿本氏による研究 [10] ではコミュニケーションの文脈とメッセージの送信方向の 2 つの要素の基づいて会話を分類している。この研究では 2 者間での会話を中心に細分化し、3 者以上の間で双方向にメッセージを送信するインタラクティブな会話は“Multi-way”モデルとして 1 つに集約している。

しかし、日常生活における会話のコンテキスト情報を扱う場合、会話が 2 者間で双方向に進行するか、もしくは 3 者以上で双方向に進行するかについて会話を区分することは効果的ではない。そこで本研究では日常生活内における会話をより細分化するため、2 者以上の人々の間で双方向に進行する会話を対象に以下の 2 つの要因に基づいて会話を分類する。すなわち会話内の話者と聴者を総合した会話参加者数、会話を行う場所と時間が参加者間で事前に定められるかを示す会話突発性である。本論文で分類した会話についてその概念図を図 2.1(a), 2.1(b) に示す。

- ミーティング型会話

少、中人数の人々が特定の場所、時刻に集合して行う会話を本論文では「**ミーティング型会話**」と定義する。ミーティング型会話には、研究会で週に 1 度開かれる定例ミーティングや企業で行われる打ち合わせ等が挙げられる。定期的に行われる、会話参加者内で事前に会話を行う時間と場所が決定されているという点で、会話突発性は低い。

- アドホック型会話

少人数の人々が事前に集合する場所、時刻を指定しなくても自然発生する会話を「**アドホック型会話**」と定義する。アドホック型会話の例としては、知人との懇談、オフィスの廊下での立ち話、研究室内で

突発する研究相談，ディスカッション，ブレインストーミングなどが挙げられる。アドホック型会話は，ミーティング型会話よりも少人数の人々が会話に参加する。またミーティング型会話のように参加者間で集合する場所，時刻が取り決められておらず，いつでもどこで発生するかわからない，非常に会話突発性の高い会話形体である。

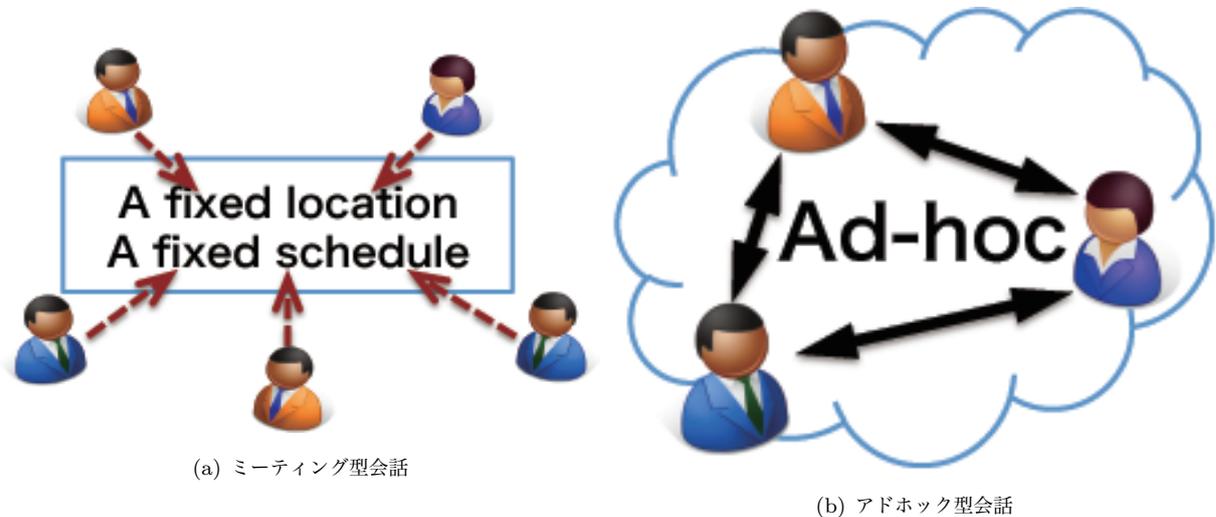


図 2.1 本論文が定義する会話形体の模式図

2.3.1 日常生活内における会話分類調査

これらの会話形体がそれぞれ日常生活内においてどの程度発生するのか調査するため，日常生活内で発生する会話に関するアンケート調査を行った。

実験手順

実験では被験者が他者と会話を行う度に指定されたアンケート用紙の質問事項に回答するという調査方法を採用した。アンケートは会話参加者の人数，会話を行なっていた時間などについて記入させる質問事項を設けた。これは会話形体分類時の分類要素に基づいて作成した。アンケートの質問事項の一部を表 2.1 に示す。このアンケート用紙の束を被験者に複数渡した上で，実験期間中常に携帯して他者と会話を行う度に 1 枚アンケート用紙に解答するよう指示した。被験者は情報工学を専攻する 19 歳から 24 歳の男性 6 人，女性 4 人の合計 10 人の学生が参加し，実験期間は 2011 年 7 月 8 日から同 14 日の間で実施した。

実験結果

実験で得られたアンケート結果をグラフ化したものを図 2.2，2.3 に示す。

図 2.2 は得られたアンケート結果のうち，各会話を参加者数と会話突発性に基づいて区分し，その発生件数をグラフ化したものである。グラフ内横軸は会話突発性を示しており，数値が小さくなるほど突発性が高いことを示している。また縦軸が会話の発生件数を示している。グラフから分かる通り，会話突発性が高く少人数

質問事項	解答項目
Q1. 会話に参加していた人数は何人ですか.	自由記述
Q2. 会話を続けていた時間を選択してください	1. 10分以内 2. 10分～30分 3. 30分～1時間 4. 1時間～3時間 5. 3時間以上
Q3. 事前に予定されていた会話ですか.	1. 全く予定されてなく突発的に起こった 2. 主観的には予定されていなかった 3. 主観的には予定されていた 4. 参加者同士で会話を始める日時場所が予定されていた
Q4. 会話の内容を簡潔に記述して下さい	自由記述

表 2.1 アンケート質問項目

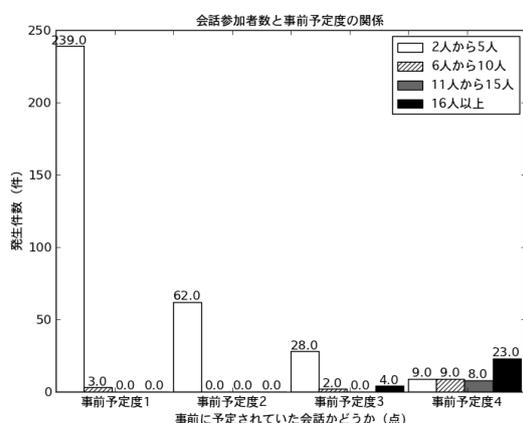


図 2.2 調査結果:会話参加者数と会話突発性

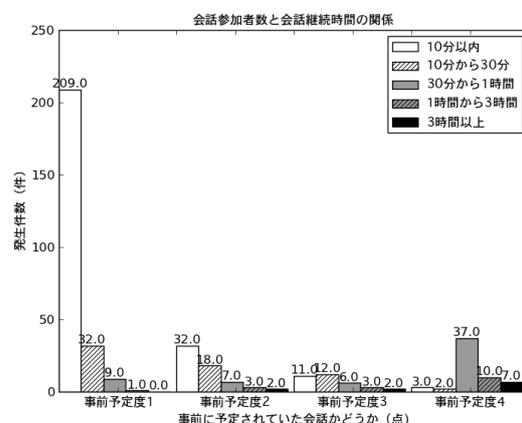


図 2.3 調査結果:会話時間と会話突発性

の参加者で会話グループが構成される, アドホック型会話にあたる回答が発生した会話の大部分占めた. 対して, ミーティング型会話にあたる会話は少数だった. これについて定例ミーティングのような集会が週に1度の周期で行うことが基準とされることや, 会話自体が長時間を要する等の理由から実験期間内において発生件数が低くなったと考えられる.

また, 図 2.3 は会話が継続した時間と会話突発性で調査結果を区分して各発生件数をグラフ化したものである. 突発性が高い会話は 10 分以内に会話が終了するケースが圧倒的に多数を占めた. この短時間で終了する会話の発生件数は会話突発性が低下するに連れて徐々に減少していく. さらにそれに伴って会話が長時間継続した会話が増加した. したがって, アドホック型会話は会話時間が短く, 突発性の低いミーティング型会話は長時間に渡り会話が継続すると言える.

自由記述形式で会話内容について回答させる設問から得られた意見として, アドホック型会話の会話内容としてとりわけ目立ったのが知人相手の雑談だったり, 研究室での研究相談であった. これは被験者全員が同

じ研究会に所属していることに起因している。ミーティング型会話の具体例としては、研究会の定期ミーティング、大学の授業でのグループワーク等話し合いの場における会話であった。これはミーティング型会話が他の会話体系より発生件数が少ないのは、定例ミーティングが週に1度のみ行われることなどの理由が考えられる。さらに大規模な会話として大学の大教室で行われる授業、サークルや体育会の全体ミーティング等も回答に含まれていた。本調査の被験者は全員学生であったため、大学内での会話が多く占めたが、他の被験者を起用して実験したとしても会話内容は異なっても、発生件数割合に大きな変化は無いと考えられる。

2.3.2 本論文の対象領域

提示した会話形態の内、本論文ではアドホック型会話を対象とした会話コンテキストの抽出を行う。実験結果が示すように、アドホック型会話は日常生活で発生する会話の大部分を占めるためである。日常生活内で頻発するイベントに関するコンテキスト情報は、コンテキストアウェアシステムによって利用される機会も多いと言える。そのため、日常生活内の会話において頻発するアドホック型会話のコンテキスト情報は、多くのコンテキストアウェアシステムで応用利用されることが期待できるため、コンテキスト情報としての利用価値が高いと言える。

さらに、本論文が想定する会話コンテキストを利用するアプリケーションはアドホック型会話の中で利用されることが好ましい。人々が会話を交わしながら利用することを想定したシステムは既存研究で提案されている [11] [12]。その中でも本研究が想定するコンテキストアウェアシステムはいつどこ起こるか分からない、日常生活内で自然発生するマイクロな会話における会話コンテキストが必要となる。ミーティング型会話は会話参加者が限定されて且つ事前予測も可能であるため、シナリオのようにサービスの対象を動的な選択の必要性は低い。同様に、これらの会話は会話を行う場所、時刻が参加者間で共有されているため、人々には会話した時間帯や場所がアドホック型会話と比較して印象づけられている。そのためシナリオのような会話コンテキストによる仮想コンテンツのアノテーション付加のようなサービスは、会話の時間や場所が限定できないアドホック会話において利用価値が高い。

このように、アドホック型会話は日常生活において多数発生する傾向にあり、本研究が想定するシナリオではアドホック型会話のコンテキスト情報が必要であるため、本研究ではアドホック型会話を対象とする。

2.4 目的と課題の整理

本研究ではアドホック型会話に関する、“いつ、どこで、誰が会話を行なったか”という会話コンテキストの抽出に焦点を当て、アドホック型会話の参加者のグルーピングを行うことを目的とする。この目的に対する解決すべき課題を以下で整理する。

2.4.1 センサ情報常時取得問題

アドホック型会話の発生は事前予測することができない。したがって、ユーザ周囲でグルーピングのために必要なセンサ情報を常時取得し続ける必要がある。

ミーティング型会話のように会話が発生しやすい場所が既知であれば、会話が頻繁に発生する環境へセンサネットワークを敷設し、それから得られる情報を利用することで会話の検知が可能である。例えば、オフィスの会議室の椅子やテーブルに加速度や照度を取得するセンサネットワークを敷設したり、会議室のカードキーロックシステムから情報を取得することによって、誰がいつ会議室内で会話を行ったか判別することがで

きる。また、参加者間で事前に会話を始める時刻が共有されていれば、ユーザの所有するスケジューラアプリケーションといったソフトウェアから得られるパーソナル情報に基づいて会話の検知を行うことも可能である。

しかし、アドホック型会話では環境内に設置した情報取得設備からの情報やユーザのパーソナル情報に基づいたグルーピングは困難である。したがって、環境内に敷設した情報取得設備や事前登録された個人情報に基づいたグルーピングは不適切であると言える。

2.4.2 同一空間内非参加者問題

アドホック型会話は会話参加者が少数であり、尚且つ突然発生する。そのため、ミーティング型会話のように1つの空間に発生する会話グループを1つに限定することができない。ミーティング型会話の例として研究室の定例ミーティングを挙げるとすると、ミーティングの参加者は研究室の会議スペースといった特定の場所に定時集合する。この時、ミーティング参加者によって会議スペースが埋め尽くされ、ミーティングに無関係の人間は同一空間内に存在しにくい。この点から会議スペースのような空間に同席している人々は会話グループの参加者であるというような、ユーザ同士の位置関係や距離を利用したグルーピングが可能である。

それに対して、アドホック型会話は会話グループのすぐ近傍に会話とは無関係の人々が存在するケースが頻繁に生じる。例えば研究室内での懇談している人々の隣に会話に加わらず作業に没頭している学生がいる、といった場合である。このような状況でユーザの位置や距離に応じてグルーピングを行う手法ではこの近傍に存在する会話の非参加者をグループの参加者であると誤認識する。本来会話に参加していない人間を会話参加者であると検知することはコンテキストを利用するアプリケーションの誤動作を引き起こす。したがって、アドホック型会話のグルーピングではグループの近傍にある非参加者を区別し、誤認識を防ぐ機構が必要となる。

2.5 機能要件

アドホック型会話のグルーピングにおける以上の課題を踏まえた上で、会話コンテキスト処理系の機能要件を説明する。

2.5.1 自然発生する会話の検知

センサ情報常時取得問題の原因は、突発的に生じる会話検知とグルーピングのためシステムはユーザ周囲から継続的に情報取得を行う必要があることである。この原因により、環境敷設型設備による情報収集ではアドホック型会話のグルーピングには不十分である。したがって、この課題を解決するためには、ユーザが常時携帯可能なセンシング端末を利用したウェアラブルセンシングによるセンサ情報の常時取得を行う必要がある。この際、日常生活内でウェアラブル端末による常時情報取得を行う場合、ユーザが端末を身に付けることによる負担も考慮しなくてはならない。グルーピングを目的とした特殊な専用端末の使用はユーザが普段携帯する端末に、本来不要な端末を1つ増やすこととなり、ユーザの負担となる。このことから、ユーザが普段から携帯する端末をグルーピングのセンシング端末として流用できることが好ましい。

2.5.2 非参加者の排除

同一空間内非参加者問題の焦点はアドホック型会話のグループを空間的に区分することができず、ユーザの位置情報や近接情報に基づいたグルーピングが有効でないという点である。位置情報や近傍情報に基づくグルーピングでは、さらに同一空間内に存在するユーザの中から実際の会話参加者を会話毎に区別する必要がある。そのため空間的なグルーピングと併用して、会話時の人間の行動特性に基づいた基準で会話参加者と非参加者を区分する機構が必要となる。具体的には各ユーザの音声情報や、会話中のユーザの挙動や行動をセンシングした情報の利用である。これらの情報に基づいてさらに細かく参加者を区分することで、グループ周囲の非参加者をグループから排除することが必要である。

2.6 本章のまとめ

本章ではまず、会話コンテキストの利用シナリオについて示した。次に本論文においてグルーピングの対象とする会話形体を明確にするため、日常生活内における会話の分類を定義した。さらに本論文の定義を補完するために会話に関するアンケート調査を行い、結果について考察を行った。これらに基づいて機能要件を示した。

第3章

会話応答頻度情報によるユーザグルーピング

2章で説明した機能要件に基づいて、アドホック型会話においてグルーピングを行う手法設計を行った。本章では本研究が提案するグルーピング手法について説明する。さらに既存研究との差分を比較する。

3.1 Milvus

本研究ではアドホック型会話におけるグルーピングを行う処理系として各会話参加者の会話応答頻度に応じてグルーピングを行う Milvus を提案する。下記に Milvus のアプローチを前章で述べた機能要件に関連付けて説明する。

3.1.1 アプローチの詳細

前章で定義した機能要件を解決するアプローチをそれぞれ説明する。まず、日常的端末で収集した情報によるコンテキスト解析を行うことで、自然発生する会話の検知に対応する。また、会話応答頻度に基づいたユーザグルーピングを実施することによって、会話グループ近傍の非参加者の排除を可能にする。

日常的端末で収集した情報によるコンテキスト解析

Milvus では、自然発生する会話の検知のために、スマートフォンによって常時収集したセンサ情報を比較的短い単位時間毎にコンテキストの解析を行う。短周期解析アーキテクチャを利用する。センサと高速演算機器、高速ネットワーク機器を搭載するスマートフォンは近年急速に普及し、センシング端末として人々の日常生活に浸透している。このスマートフォンをセンシング端末として利用することで、ユーザが日常生活に本来必要のない会話コンテキスト解析用特殊デバイスを携帯する必要を取り払うことができる。

会話応答頻度に基づくユーザグルーピング

近傍の非参加者を考慮しながら会話グループの解析を行う手法として、本研究では“単位時間内において誰が誰の発言に何度応答したか”を表す会話応答頻度に基づくグルーピング手法を提案する。アドホック型会話は会話中に話者と聴者が入れ替わる話者転換が頻繁に発生する。そのため特定の参加者が単独で長時間話し続けるという状況は発生しづらく、ある参加者の発言に対し他の参加者が応答するという形式で会話が進行す

る。本研究ではある話者に対して頻繁に応答を行った人物は互いに同じ会話に参加していると仮定し、これを利用してグルーピングを行う。どのユーザが誰と会話応答を頻繁に行ったかという指標に基づいてグルーピングすることで、既存手法で提案されるような、単純に人々の位置情報や近接情報に基づいて人々の群を抽出するような手法と比較してより近傍の非参加者を考慮したグルーピングが期待できる。

3.1.2 手法のデザイン

上記2つのアプローチを採用する上で、まず前提としたのはアプローチの1つとして説明した、センシングに使用する日常的な端末としてスマートフォンを利用するという点である。日常的に普及した携帯端末の中でスマートフォンがセンシング端末として有用であることは上述した通りである。この上で、位置情報や近接情報といった空間的情報の他にアドホック型会話におけるユーザグルーピングに有用なセンサ情報について考える。スマートフォンで取得可能なセンサ情報の中で会話と密接に関係する情報は音声情報であると言える。スマートフォンのマイクで音声情報を常時取得することで、端末を携帯するユーザの会話音声を取得することができる。この会話音声に基づいてグルーピングを行うことで、実世界で行われる会話を直接的にグルーピングへ反映することが期待できる。

会話音声に基づいてグルーピングを行う場合、アドホック型会話においては人々の会話上の応答関係を追うことによって、誰と誰が会話を行なっているか判別できると考えられる。なぜなら、前章で詳述したように、アドホック型会話は会話内で単一の参加者が一方的に話し続けるのではなく、会話参加者が双方向的に発話を行うためである。互いに話者と聴者が入れ替わって会話が進行するため、この入れ替わりが頻繁に発生する人々は同じ会話へ参加していると考えられる。したがって、本研究ではこのような会話応答関係が得られれば、各ユーザの会話相手が判別でき、会話グループが取得可能である、という仮説を立てた。

人々の会話応答関係を音声情報から把握する手法は複数考えられるが、本研究では話者と聴者が入れ替わる頻度をグルーピングに用いる。会話の進行状況を把握する方法として、音声情報を自然言語解析することで得られる会話トピックからユーザ間での会話遷移の把握などが考えられる。しかし、自然言語処理は、スマートフォン上で音声処理を行うことや一度に複数の音声情報を処理しなければならないことを考慮すると、非常に処理コストがかかると考えられる。そこで本研究では自然言語処理を行わずに取得可能な会話応答関係を判別する指標として会話応答頻度を利用する。会話応答頻度はユーザが発話を行った時系列の入力区間に限って取得し、それらと比較することで得られる話者と聴者の転換頻度である。これに基づいてグルーピングを行うことにより、自然言語処理を行う場合と比較して人々の会話応答関係を低コストで利用しながら、会話応答関係に基づくグルーピングが期待できる。

以上の考察から、上述したアプローチを選択し、アドホック型会話のユーザグルーピングを行うこととした。

3.1.3 想定環境

Milvus はシステム動作環境として、まずセンシング端末として高い演算能力と高速なネットワーク、GPSといった各種センサを内蔵したスマートフォンをユーザが各自携帯しており、常時スマートフォン上でセンサ情報を取得するアプリケーションが動作していることを想定している。Milvus がグルーピングに使用するセンサ情報は GPS から得られる位置情報とマイクから得られる音声情報であるため、アプリケーションは常時端末の位置情報と音声の離散情報を取得し続ける。さらにユーザが会話時に発する音声を取得しやすいように端末には外部集音マイクが取り付けられた上に集音マイクが胸元に露出している等、ユーザの音声を取得しや

すい状態で携帯されているとする。スマートフォンは端末 ID とユーザの個人情報が関連付けられており、端末 ID を通じてユーザを一意に決定できるとする。

Milvus においてスマートフォン上で行う処理は情報収集処理のみに留める。これは一般的な端末を流用するという点で端末のバッテリー消費への配慮と演算能力を考慮した結果である。そのため Milvus では別途グルーピング処理を行う遠隔サーバが用意されていることを前提とする。遠隔サーバでは各スマートフォンから定期的送信される基礎データを受信した上、それらの情報に基づいて短期定期的に会話グループの検知と参加者の判別を行う処理が実行される。さらに解析結果は遠隔サーバが参照可能なストレージに蓄積されることとする。



(a) 大学棟内共同スペース

(b) 研究室オフィス

図 3.1 セミパブリックな空間の例

Milvus はアドホック型会話に対するグルーピング処理を提供する。本研究ではとりわけ大学構内の研究室のようなセミパブリックな空間において知人同士の間で自然に発生する会話を対象とした。これは本研究が想定するコンテキストウェアシステムが頻繁に利用されると考えられる空間であること、さらに前述した実験においてアドホック型会話が発生しやすいという知見が得られたことを考慮したためである。このセミパブリックな空間とは互いに知人である人々が共同で利用する空間のことを指し、全く見ず知らずの人々だけで共同利用するパブリックな空間とは異なる。図 3.1(a), 3.1(b) は大学構内のセミパブリックな空間の一例として挙げられる、共同談話・休憩スペースと研究室オフィスの光景である。本研究ではこのような空間において発生する会話を対象とする。したがって、Milvus の細かいパラメータや予備実験の結果はこのセミパブリック空間内における知人同士の会話に対して若干のバイアスがかかっていると考えられる。ただし街中で見ず知らず相手と思いがけず交わすような会話であっても、パラメータのチューニング次第で柔軟に対応可能であるようにシステムは設計されている。

3.1.4 処理の流れ

Milvus によるグルーピング処理の流れについて述べる。Milvus は“誰が誰の発言に対して何回受け答えをしたか”というユーザ間の会話応答頻度に基づいてアドホック型会話の参加者グルーピングを行う。Milvus のグルーピング処理は下記のステップに区別される。

1. 話者認識処理による発話区間の取得

スマートフォンを携帯するユーザが単位時間内においてどのタイミングでどれだけ発話したかを表す発

話区間を算出する。発話区間はユーザが発話を行う毎に記録された発話開始時と発話終了時のタイムスタンプで構成された時刻情報である。発話区間の算出はスマートフォンで取得した音声を話者認識アルゴリズムに基づいて解析する事によって行う。これによって得られる単位時間内の全発話区間をその後の処理に利用する。

2. 会話応答頻度算出対象の絞り込み

会話応答頻度を算出する際に発話区間情報を比較する対象ユーザ群を各ユーザの位置情報と音声情報に基づいて絞り込みを行う。絞り込みは会話グループ内で積極的に会話を継続するグループの**センタユーザ**の選出と各センタユーザを軸として会話応答頻度算出対象となる**周辺ユーザ**の限定によって行う。

3. 会話応答頻度の算出

センタユーザと周辺ユーザの単位時間における発話区間情報を比較していき、センタユーザに対する各周辺ユーザの会話応答頻度を算出する。発話区間情報の比較は時系列を単純比較することで行う。

4. 会話応答頻度情報に基づいたユーザクラスタリング

会話応答頻度情報の値に基づき、センタユーザと周辺ユーザの群で形成される群にユーザをクラスタリングする。この処理によって得られるユーザクラスタを会話グループとして、それらの集合を会話コンテキストとする。

以上の手順で会話グループの検出を行う。次節より各手順に関する詳細な処理について説明する。

3.2 発話区間の取得

スマートフォンのマイクから取得できる音声の離散波形データを基礎データとして、単位時間内で各ユーザが発話した時系列区間を求める。個々の発話区間は発話開始時刻と発話終了時刻で構成される。あるユーザが単位時間内に他者と会話を行った際、ユーザが言葉を発する度に発話を開始した時刻と終了した時刻を記録し、発話区間とする。単位時間内においてあるユーザの発話区間を全て検出して、その集合を単位時間内におけるユーザの発話区間情報とする。

マイクから取得した音声のオンライン解析による発話区間検出は、音声解析分野において盛んに手法研究が行われている。波形の振幅と零交差数に基づく発話区間検出 [13][14] や、音声情報からメル周波数ケプストラム (MFCC) を抽出し、そのケプストラム係数を特徴量として使用した混合ガウス分布モデルに基づく話者認識手法 [15][16] 等の手法が代表的である。本研究は新たな発話区間検出アルゴリズムの提案を対象としているわけではなく、アドホック型会話における参加者グルーピングに焦点を当てている。したがって、発話区間検出には既存のアルゴリズムを活用することで実現する。

このように全てのユーザに対して単位時間内において検出した発話区間情報を会話応答頻度の算出やグルーピング処理に用いる。

3.3 会話応答頻度の算出

前節で言及した各ユーザの発話区間情報を基に会話応答頻度を算出する。本節では会話応答頻度の詳細に言及した上で、算出時のエラー低減を目的とした算出対象の絞り込み処理、及び会話応答頻度の算出処理に関して説明する。

3.3.1 会話応答頻度

アドホック型会話は会話参加者の間で話者と聴者が入れ替わりながら進行する。例えば知人同士の懇談では特定の参加者が一方的に話し続けるという場面と比較して、全参加者が話し役と聞き役の両方をこなして多方向的なコミュニケーションが行われる傾向が強い。これはある参加者の発言に対して別の参加者が反応し、それに対して受け答えを行う、というやり取りの繰り返しによってアドホック型会話は構成されているとも言える。

このような会話内において発言を行った話者とそれに対して応答した人物は互いに同じ会話に参加していると考えられる。そこで本研究ではあるユーザの発言に対して頻繁に応答を行うユーザは発言したユーザと同じ会話グループに属しているという仮説を立てた。この仮説に基づき、互いの発言へ頻繁に応答を行うユーザ同士の集合を会話グループとして判定するという方針の下、ユーザ間の会話応答頻度に基づくグルーピング処理の構築を行った。

会話応答頻度は単位時間内においてあるユーザの発言に周囲のユーザがそれぞれ応答をした回数である。本研究ではこの会話応答頻度を自然言語処理ではなく、発話区間を時系列的に比較することで算出する。具体的にはあるユーザの発話に対して発話開始直前と発話終了直後に発言したそれぞれのユーザをあるユーザに対する応答者であるとする。すなわちあるユーザが発話を開始した時刻に最も近い時刻で発話を終了した、もしくはあるユーザが発話を終了した時刻に最も近い時刻で発話を開始したユーザを、あるユーザと互いに応答していると判定する。そしてあるユーザの単位時間内における全発話に対して応答者となった回数があるユーザに対する会話応答頻度情報とする。

図 3.3 に発話者と応答者の時系列的な遷移関係を図式化したものを示す。図中の user は会話参加者を示しており、矩形は user が単位時間中に発話した区間を示している。図中の横軸は時系列を表しており、図中右に進むほど時系列が進むことを意味する。そして各 user の発話が終了した時刻に最も近い時刻に発話を開始した user をその発話に対する応答者とする。図中の破線は各 user の発話応答の関係を示している。本研究ではこのように会話内の発話者の応答関係が成っているとし、各発話の開始、終了時刻を比較することで会話応答頻度を算出する。

会話グループの検出に会話応答頻度を用いる事によって、ユーザの位置情報や近接情報に基づいてグルーピングを行う手法と比較して、会話グループの近傍に存在する無関係なユーザを会話参加者として誤検出することを低減できることが期待できる。なぜなら、互いに近接し合っているユーザの中から、会話状況に基づいてさらにユーザを区分する事ができるためである。会話応答頻度から会話参加者の発話に反応しない、もしくは全く別の会話に参加しているという状況を把握することが出来、それに基づいて近傍する会話非参加者をグループから排除できると期待できる。以上の様な利点を考慮して、本研究では会話応答頻度に基づいてグルーピングを行う。

会話応答頻度の算出において、音声情報を自然言語解析することで各発話の会話内容を把握し、会話内容の

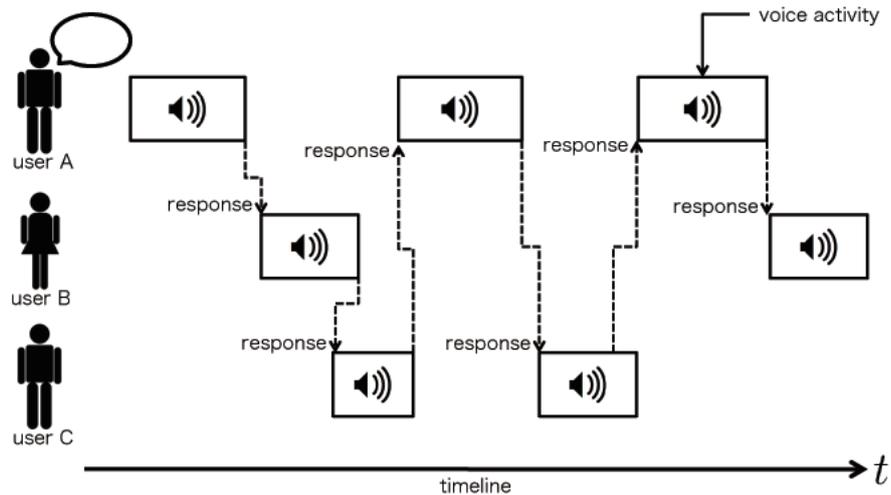


図 3.2 発話者と応答者の時系列的な遷移

繋がりに基づいて応答相手を特定する方法は算出手法の 1 つとして数えることができる。しかし、自然言語処理による音声情報の発話内容解析をリアルタイムに行うには、バッテリー消費量と演算速度の観点よりスマートフォンは実行端末として不十分である。またスマートフォンで取得した音声情報を遠隔のサーバにストリーミングして音声解析処理を計算資源が豊富なサーバ上で行うという方法も考えられるが、日常生活内の会話音声を送信する点において、ユーザのプライバシーが損なわれてしまう。以上の理由から本研究では音声自然言語処理に基づく会話応答頻度算出は対象としない。

3.3.2 対象ユーザの絞り込み

会話応答頻度を時系列的基準のみに従って算出する場合、算出結果と実際の応答者に食い違いが発生する恐れがある。具体的にはあるユーザの発話に対する実際の応答者の応答と、別グループの発話者の発話が時系列的に入れ替わり、結果的に別の発話者が応答者として判定してしまう事例である。図 3.3 は応答者の食い違いが発生する場合を図示したものである。図中の userA, userB, userC の集団と userD, userE の集団はそれぞれ同じ会話グループに属している。図 3.3 と同様、各 user の発話に対する応答関係は破線で示されている。図中赤破線では userA の発話の応答者が本来の応答者である userB ではなく userD であると食い違って判別されていることを示している。異なる会話グループに属する userD の発話が userA の発話終了時刻に最も近い時刻に開始されたため、その後に発話が開始された userB ではなく userD が userA の発話に対する応答者であると判別されてしまう。このように発話の開始、終了時刻の比較のみでは応答者の食い違いが起り、本来の会話応答頻度の正確な算出を妨げてしまう。

そこで本研究では、会話応答頻度情報の算出範囲を限定することで、応答者の食い違いの発生を低減させる。低減手法として、位置情報による粗粒度ユーザフィルタリング、及び単位時間内の発話区間情報を処理することで得られる高頻度発話者と同時発話者に関する情報に基づいた、会話応答頻度算出対象の分類によって、算出範囲の限定を行う。

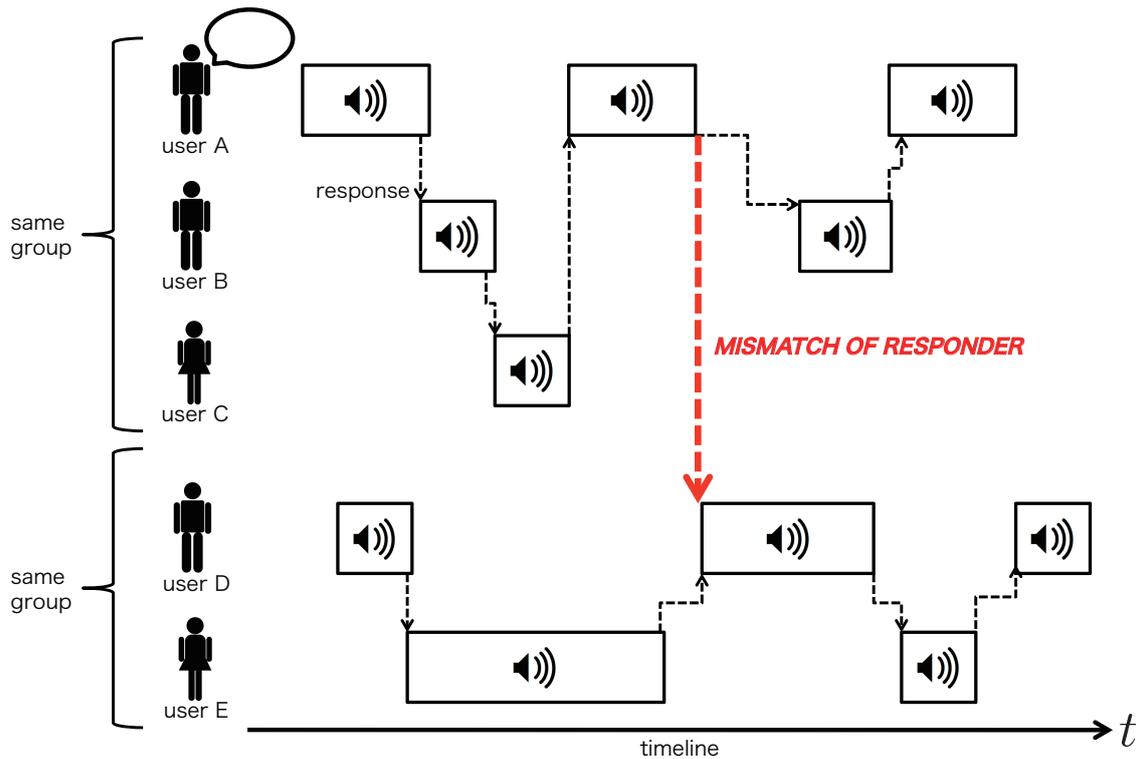


図 3.3 会話応答者の誤検知

位置情報に基づく粗粒度フィルタリング

グルーピングを行う全ユーザを同一に扱って会話応答頻度を算出することを考えた場合、大量の発話者が同時発生する状況が必然的に起こるため、応答者の食い違いの発生頻度が高まる。一方で全ユーザの単位時間内における発話区間の比較は、ユーザの増加に合わせてグルーピング処理の計算量が爆発的に増大する。以上の理由から、会話応答頻度算出以前にユーザをある程度区分する必要がある。

そこで Milvus では会話応答頻度の算出処理を行う前に、ユーザの位置情報に基づいてユーザを粗い粒度でグルーピングする。アドホック型会話は空間的な指標のみで参加者グルーピングを達成することは困難である。しかし、空間的に全く近傍しないユーザ同士が対面で会話を行なう状況は考えづらいため、空間的に近傍するユーザ同士は同じ会話グループに属している可能性が高いと言える。そのため、位置情報が近傍するユーザの集団の中で発話区間の比較を行うことにより、応答者の食い違いと計算量を抑えることが期待できる。Milvus ではユーザが所持するスマートフォンの GPS といった位置測位技術から得られるユーザ位置情報に基づいてユーザ間距離計算を行い、互いに近距離にいるユーザの集団を生成する。このユーザの集団に対して会話応答頻度を基にしたグルーピング処理を施す。

高頻度発話者の選出と会話応答頻度算出軸の設定

位置情報に基づいたユーザのフィルタリングにより会話応答頻度の算出対象を絞った上で、Milvus ではさらに発話区間情報を用いた事前処理によって会話応答頻度算出対象の限定を行い、会話応答頻度算出の信頼性

向上を目指す。

発話区間の事前処理による算出対象の限定はまず、単位時間内における高頻度発話者の選出を行う。高頻度発話者とは単位時間内において発話を行う頻度が高いユーザであり、周囲のユーザに比べて単位時間に対して発話時間が占める割合が多いユーザを指す。図 3.4 は図中の userA が他の user と比較して単位時間 t 内における発話区間が多いことを示し、結果として userA が高頻度発話者として選出されることを図示している。このように発話区間の総時間が単位時間を占める割合が高いユーザを高頻度発話者とする。

Milvus では、この高頻度発話者を中心として会話応答頻度の算出を行う。具体的にはユーザに対して総当たりで会話応答頻度を算出するのではなく、高頻度発話者の発話に対する応答についてのみ会話応答頻度の算出を行う。これによって応答者の食い違いを抑え、会話応答頻度の効率的な算出が期待できる。

高頻度発話者を軸とした会話応答頻度の算出は、アドホック型会話は会話参加者間で話者と聴者が入れ替わりながら進行する傍ら、グループ内の特定の人物が率先的に発話を行なって会話を主導する、という仮説に基づいている。例えば、少人数によるディスカッショングループが自然発生した場合には参加者の誰かが会話の主導権を握って議論を進行するということが考えられる。また、日常生活内における懇談に関しても、参加者の一人が話し続け、他の参加者がそれに相槌を打つような光景が見られる。これに関して、Bavelas 氏による小グループ内における会話の傾向に関する実験と考察結果 [8] では、小規模の会話グループに対して簡単な課題を与え、グループ参加者が互いにコミュニケーションをしながら課題解決を行った時のコミュニケーションの流れを記録するという実験を行なっている。その結果、他の参加者の意見が集約されるコミュニケーションのハブ役が発生するように議論を行ったグループはハブ役のないグループに比べて与えられた課題に対するエラーが少なくなるという知見が得られている。さらに、ハブ役の有無にかかわらず議論の進行を主導するリーダー役の自然発生が確認されたという結果が得られた。これらを踏まえて、本研究では会話を主導するユーザは単位時間内で他のユーザに比べて発話する頻度が高いとし、この高頻度発話者が会話グループの中心となっていると考える。そして、高頻度発話者に対する会話応答頻度を算出し、グルーピングに用いることとする。

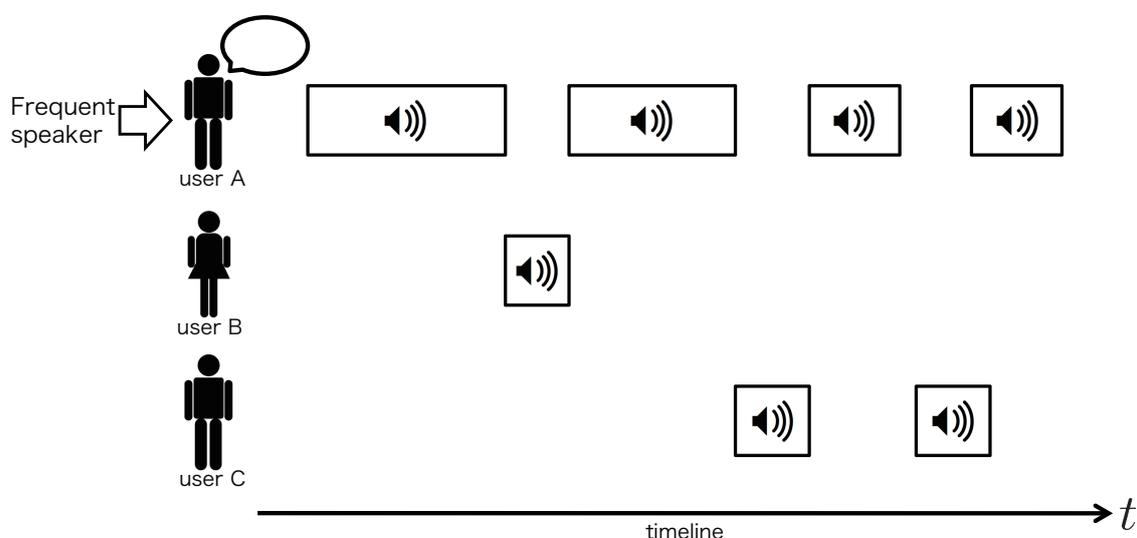


図 3.4 高頻度発話者

ある単位時間における高頻度発話者の選出はアルゴリズム 1 に示すアルゴリズムによって行う。このアルゴ

Algorithm 1 高頻度話者の選出

Require: $U = \{u_1, u_2, \dots, u_i\}$, $\mathbf{S} = \{S_{u_1}, S_{u_2}, \dots, S_{u_i} | S = \{s_1, s_2, \dots, s_j\}\}$ **Ensure:** $L = \{l_1, l_2, \dots, l_k\}$

```
1: function lead_speakers( $U, \mathbf{S}$ ):
2:   for  $u$  in  $U$  do
3:     for  $s_{ju}$  in  $S_u$  do
4:       get time  $a_j$  which  $u$  spoke in  $s_{ju}$    comment: synonymous with  $a_{ju} \leftarrow |s_{ju}(\text{end}) - s_{ju}(\text{start})|$ 
5:     end for
6:     store the sum of  $u$ 's all speaking time spans  $\sum_{j'=0}^j (a_{j'u})$  to  $\dot{a}_u$ 
7:     add  $\dot{a}_u$  to  $A$ 
8:   end for
9:    $L \leftarrow \text{divide}(U, A)$ 
10: return  $L$ 
11: end function
12:
13: function divide( $U, A$ ):
14:   calculate  $\bar{x}$ , the average of  $A$ 
15:   for the total  $a$  of  $u, \dot{a}_u$  in  $A$  do
16:     if  $\dot{a}_u \geq \bar{x}$  then
17:       add user  $u$  to  $L$ 
18:     end if
19:   end for
20: return  $L$ 
21: end function
```

リズムは単位時間内の各ユーザ u が行った発話 S_u を構成する各発話 s_{ju} の発話時間 a_j の合計 \dot{a}_u を算出して、その合計発話時間に従ってユーザの集合 U を2つに分割する関数 `lead_speakers` を示している。ユーザのある発話 s_{ju} は2つのタイムスタンプで構成される集合であり、第一要素 $s_{ju}(\text{start})$ はユーザ u が発話を開始した時刻を表し、第二要素 $s_{ju}(\text{end})$ は終了した時刻を表す。`lead_speakers` 内では U の各ユーザに対してユーザの発話時間の合計を計算し、クラスタリング関数 `divide` へ渡す。`divide` 関数は単位時間あたりの発話時間がその平均より上回っているユーザの集合を返す関数である。最終的に発話時間が平均を上回るユーザの集合 L を高頻度発話者として選出する。

このアルゴリズムによって会話応答頻度算出の母集団となるユーザの中でも、単位時間内において他のユーザより頻繁に発話を行ったユーザを選出できる。選出基準に合計発話時間の平均値という可変閾値をとることによって、会話グループ間の発話量の違いを吸収することができ、固定閾値による選出よりも柔軟な選出が期待できる。

このようにしてユーザの中から高頻度発話者の集団を選出し、会話応答頻度算出時の軸として利用する。

発話区間のオーバーラップ

この一方で各高頻度発話者と応答を行う他ユーザの関連性について考える。アドホック型会話の会話グループが発生した周囲に別の会話グループが同時並行的に形成され、しばしば会話グループの近くに別の会話グループが存在するという状況が発生する。このような状況下では高頻度発話者に対する応答が同時並行発生し、応答者の食い違いを誘発する。会話グループの周囲に他のグループが存在しないような場合であっても、グループに近接する非参加ユーザの独り言を発話者に対する応答と誤判定することも考えられる。したがって、高頻度発話者の選定だけでなく、高頻度発話者に対して応答を行う周辺ユーザについても、会話応答頻度算出以前に対象範囲を限定する必要がある。

Milvus では、単位時間内において同時に発話を行うユーザ同士は互いに同じグループに属さないとし、単位時間内において発話区間が時系列的に重なる（オーバーラップする）傾向にあるユーザ間の発話応答を無視する。そして発話がオーバーラップするユーザは互いに同グループに属さないユーザであるという定義情報を生成した上で、その後の発話応答頻度算出とグルーピング処理のパラメータとして使用する。これは、別グループの会話中で行われる発話や非参加ユーザの独り言は、会話グループに属す本来の参加ユーザ同士の会話と完全に独立して行われ、本来の参加ユーザの発話に平行して起こることが予測されるためである。例えば、ある会話グループに属すユーザ間の会話は発話者の発言をある程度聞いてから別のユーザがそれに応答するという会話応答関係が起りやすい。しかしその会話グループと同時並行に会話が行われる別グループ内の発言は会話グループのそういった発言の流れを無視して行われるため、グループ間で発話者が並行して存在する状況が発生する。したがって、複数の発話者が同時に発話を続けるという状況が生まれ、その発話者同士は互いに別グループに属するということが言える。これを利用して、会話応答頻度算出時やグルーピング処理時に利用可能な、同じグループに属し得ないユーザの集団を事前情報として生成することができる。

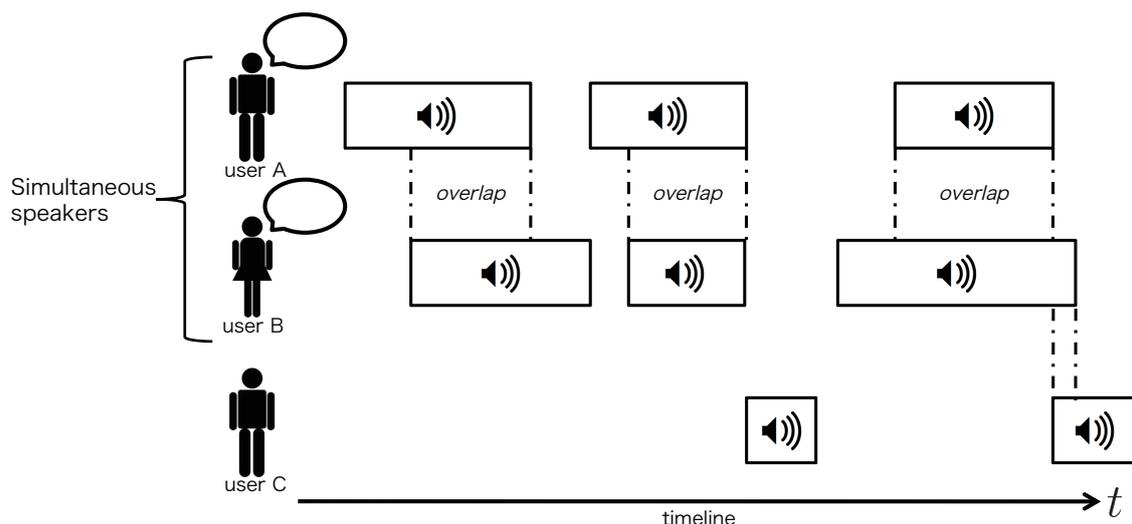


図 3.5 発話区間のオーバーラップと同時発話者の選出

図 3.5 は発話のオーバーラップが発生した様子を図に示している。図中の矩形は各 user の発話区間を示しており、破線は発話区間がオーバーラップしている箇所を表している。図中の userA と userB は他のユーザよりも互いに発話区間がオーバーラップしていることを表している。本研究では、図が示すような場合において

userA と userB は同時発話者関係にあるとし、userA と userB は同じ会話グループ内に属しないと定義する。また、図中 userC は他ユーザとのオーバーラップは確認できるものの、userA と userB のオーバーラップの総区間に比べてごく軽微である。この userC のように他の同時発話者のオーバーラップ区間と比較して、オーバーラップが少ないユーザについては、同時発話者は存在しないと判別する。このような発話区間のオーバーラップの発生を発話区間情報から事前に検知して同時発話者の組み合わせを定義し、会話応答頻度算出とグルーピング処理のパラメータとして利用する。

同時発話者の選別

発話のオーバーラップ区間から発話者と異なる会話グループに参加する、同時発話者を選び出す。これに関して、同時発話者とユーザが所属する会話グループの関係について、アドホック型会話の発話オーバーラップ量と同時発話者同士が所属する会話グループの関係性を調査する予備実験を行った。予備実験では、大学研究室内の学生の会話音声約 30 分間録音し、一定単位時間毎に、室内にいる学生の発話が周囲の学生が行った発話とオーバーラップした時間、及びその時の会話グループの形成状況を記録した。オーバーラップした時間は室内にいる学生毎に算出し、周囲の学生それぞれに対して一対一で計測した。グループの形成状況とは単位時間内における会話グループの数量と各会話グループの参加学生を指す。さらに単位時間内において学生の発話にオーバーラップした時間で降順に他の学生をソートし、オーバーラップ時間が多かった学生から順に、同じ会話グループに属しているか計測を行った。実験時に室内にいた学生は 6 人であったため、各学生とその他 5 人に対して 2 者間のオーバーラップ量が計測できた。この内、オーバーラップ時間が 0 である学生との計測結果と、5 人のオーバーラップ時間にほとんど差が見られない場合の計測結果は除外した上で、2 人の学生が同じグループに属していたか、異なるグループに属していたかを調査した。

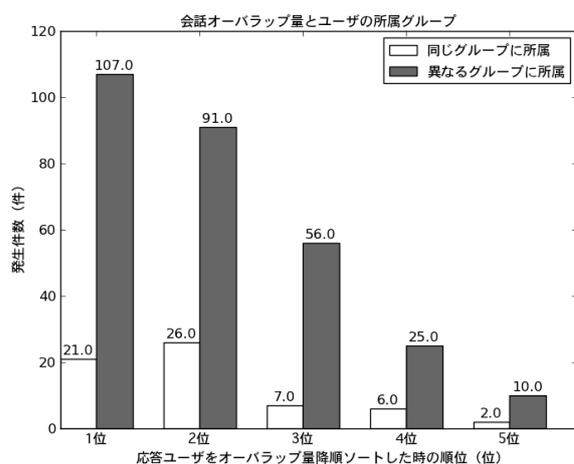


図 3.6 発話オーバーラップと所属グループ

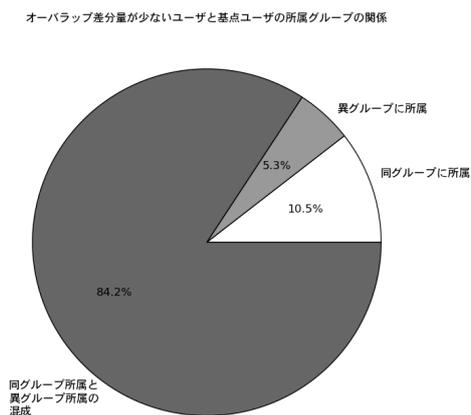


図 3.7 オーバーラップ最多者と次点者間のオーバーラップ差分が小さい場合の 3 ユーザの所属グループ

図 3.6 に調査結果を示す。横軸は各学生の発話へオーバーラップした時間が長い順に応答者にあたる学生をソートした時の順位を示し、縦軸は発話の起点となる学生とオーバーラップした学生が同じ、または異なるグ

グループに属していた場合の件数を表している。すなわち、横軸の値が1位であるグラフは、単位時間内において発話者に最もオーバーラップした周辺ユーザが発話者と同じグループに属していた場合とそうでない場合の、それぞれの発生件数を表す。

調査結果から、発話者の発話に最もオーバーラップした応答者とその次点の応答者は発話者と異なるグループに参加している事例が多いことがわかった。また、それ以降の応答者についても発話者と異なるグループに属している傾向にあったが、オーバーラップとして観測できる事例自体が少ないため、会話応答頻度算出時に用いるパラメータとしては期待できない。したがって本研究では単位時間内において発話者に最もオーバーラップした応答者と次点の応答者がその発話者の同時発話者であるとする。

ここで、ある発話者に対する最多オーバーラップ応答者と次点応答者のオーバーラップ量の差分が大きい場合を考える。最多オーバーラップ応答者のオーバーラップ量に対して次点応答者のオーバーラップ量が極端に少ない場合、最多オーバーラップ応答者は発話者と異なるグループに属していると予測できるが、次点応答者は発話者と異なるグループに属するか判別できない。なぜなら、次点応答者が発話者と同グループに属し、発話者に対する応答がオーバーラップしてしまった可能性を否定できないためである。したがって、本研究では最多オーバーラップ応答者と次点応答者のオーバーラップ量差分が大きい場合、次点応答者を同時発話者から除外する。

次点応答者が最多オーバーラップ応答者と共に同時発話者になり得る場合は、2者のオーバーラップ量がほぼ同量の場合である。これに関して、最多オーバーラップ応答者と次点応答者のオーバーラップ量の差分が少ない場合、発話者と2人の応答者はそれぞれどのように会話グループに所属するのか調査を行った。調査は前述した調査の結果の中で、最多オーバーラップ応答者と次点応答者のオーバーラップ量が1秒未満である場合に発話者と各応答者が参加した会話グループについて調べた。調査の結果、発話者と各応答者は全員同じ会話グループに参加するケース、発話者と応答者は別のグループに参加しているケース、一方の応答者は発話者と異なるグループに属していたがもう一方の応答者は同じグループに属しているケースが観測された。各ケースの発生割合をグラフ化した図を図3.7に示す。グラフから分かる通り、一方の応答者のみ発話者と同じグループに属しているケースがほとんどであった。さらに、このケースでは最多オーバーラップ応答者が発話者と異なるグループに属している場合もあれば、次点応答者のみが異なるグループに属している場合もあり、それぞれほぼ同数発生した。したがって、2人の応答者のオーバーラップ量がほぼ同量である場合はその中から発話者に対する同時発話者を選別することは難しく、むしろ最多オーバーラップ応答者が同時発話者となり得ない場合も考慮する必要があるとわかった。

以上の結果から、Milvusでは単位時間内において発話者に対するオーバーラップ量が最も多かった応答者をその発話者の同時発話者として選別する。この時、オーバーラップ量が0秒、発話者に対する全応答者のオーバーラップ量のばらつきが少量、次点の応答者のオーバーラップ量との差分が小さい場合のいずれかに当てはまる場合は、対象単位時間内においてその発話者に対する同時発話者は存在しないと判定する。これらの基準で同時発話者の選別を行う。

センタユーザの選出と周辺ユーザの限定

選出した高頻度発話者と同時発話者の情報を利用して、会話応答頻度算出軸となるセンタユーザの決定及び、高頻度発話者に応答する周辺ユーザの選別を行う。

センタユーザと共にセンタユーザ毎に周辺ユーザの選定を行う擬似コードをアルゴリズム2に示す。関数 `centers_and_targets` はセンタユーザの選定と選定したセンタユーザに対して周辺ユーザそれぞれ定義する関数である。入力値はグルーピングの対象となる i 人のユーザの集合 U 、アルゴリズム1の処理によって得られる j 人の高頻度発話者の集合 L 、各ユーザに2人定義されている同時発話者の集合 O 、そして単位時間中に全

Algorithm 2 センタユーザの選出と会話応答頻度算出対象の限定

Require: $U = \{u_1, u_2, \dots, u_i\}, L = \{l_1, l_2, \dots, l_j | l \in U\}, O = \{o_{u_1}, o_{u_2}, \dots, o_{u_i} | o \in U\}, Q = \{q_1, q_2, \dots, q_k | q \in U\}$

Ensure: $\mathbf{X} = \{X_{c_1}, X_{c_2}, \dots, X_{c_n} | X_c = \{x | x \text{ is a center user } c\text{'s target for calculating response frequency information}\}\}$

```
1: function centers_and_targets( $U, L, O, Q$ ):
2:   for all  $u$  in  $U$  do
3:     if  $L$  contains both  $u$  and  $o_u$   $\cap$  the set of center users  $C$  do not contains two then
4:       add both  $u$  and  $o_u$  to  $C$  as a center user
5:     end if
6:   end for
7:   if  $C$  has no user then
8:      $C \leftarrow L$ 
9:   end if
10:  for all  $u$  in  $U$  do
11:    if  $u$  is not in  $C \cap u$  is not in  $Q$  then
12:      add the user  $u$  to the set of marginal persons  $M$ 
13:    end if
14:  end for
15:  for all  $c$  in  $C$  do
16:    for all  $m$  in  $M$  do
17:      if  $m$  is not the simultaneous talker of  $c$ , which is defined in  $O$  then
18:        add  $m$  to the set of targets for response frequency calculation  $X_c$ 
19:      end if
20:    end for
21:    add  $X_c$  to  $\mathbf{X}$    comment:  $\mathbf{X}$  is the definitions that are constructed a center user  $c$  and targets  $X_c$ 
                       for the calculation of  $c$ ,  $X_c = \{x | x \in M\}$ 
22:  end for
23:  return  $\mathbf{X}$ 
24: end function
```

く発話区間が観測されなかった、無発話ユーザの集合 Q である。この内、 O を構成する同時発話者 o_u は前述の選定基準によってユーザによっては存在しない場合がある。centers_and_targets 関数の処理結果 \mathbf{X} はセンタユーザ c に対する周辺ユーザ x の集合 X_c の集合であり、すなわちセンタユーザに対応する会話応答頻度算出対象が集合 X という形で表される。会話応答頻度を X_c に対して算出する場合は、各センタユーザ c の発話区間を X_c 内の各周辺ユーザの発話区間と時系列的に比較することによって行う。

center_and_targets 関数はまず高頻度発話者の中からセンタユーザを選定する。2 行目から 9 行目までの処理がセンタユーザ選定処理を示している。高頻度発話者は単純に発話区間が多いユーザの集合であるため、そのまま会話応答頻度算出の軸とするには信憑性が薄い。なぜなら会話グループ内に高頻度発話者が複数存在す

る場合等を考慮できないためである。そこで Milvus では高頻度発話者の中に同時発話者の関係にあるユーザを優先的にセンタユーザとして選定する。前項の通りあるユーザに対する同時発話者はそのユーザと異なるグループに参加していると考える。このことから、高頻度発話者 l の同時発話者 o_l が L の中に同時に存在する場合、その 2 ユーザをそれぞれセンタユーザとして選定する。どの高頻度発話者も同時発話者が L に同時に存在しない場合は、高頻度発話者をセンタユーザとして扱う。

次に選定したセンタユーザの集合 C に基づき、各センタユーザ c に対する周辺ユーザを選定する。周辺ユーザは U から C と無発話ユーザの集合 Q に含まれるユーザを除いた集合 M の中から選ばれる。アルゴリズム 2 における 15 行目から 20 行目で周辺ユーザをセンタユーザ毎に選定しており、 M の中でセンタユーザと同時発話者関係が O で定義されているユーザを除外したユーザの集合がそのセンタユーザの周辺ユーザとして選定される。この処理を C の各センタユーザに対して行い、全結果を X とする。この集合 X に基づいて会話応答頻度の算出を行う。

3.3.3 会話応答頻度抽出アルゴリズム

前節で示した処理から得られるセンタユーザとその周辺ユーザの定義に基づいてセンタユーザ毎に会話応答頻度の算出を行う。アルゴリズム 3 に算出アルゴリズムを示す。response_frequency 関数は、前節で説明した処理から得られるセンタユーザの集合 C 、センタユーザ毎に定義された周辺ユーザ群の集合 X 、各ユーザの発話区間 S を入力情報とする。 S の要素である、ユーザ u の発話区間の集合 S_u で表され、 S_u を構成する各発話区間 s_{uj} は 2 つの時刻情報の集合である。初項の $s_{uj}(\text{start})$ はユーザ u の j 番目の発話が開始された時刻、次項 $s_{uj}(\text{end})$ は発話が終了した時刻を示す。そして、各センタユーザとそのセンタユーザの周辺ユーザ間における会話応答頻度 Z_c を算出し、全結果を Z として返す。

response_frequency 関数はまず各センタユーザ c の各発話区間 s_c の発話開始時刻に最も近い時刻で発話が終了した発話区間を持つユーザ、及び s_c の発話終了時刻に最も近い時刻で発話が開始した発話区間を持つユーザを U の中から選び出す。コード中 3 行目から 7 行目で s_c の発話の直前に会話していた応答者の探索を行う。すなわち発話開始時刻 $s_c(\text{start})$ と発話終了時刻の時系列的差分が最も少ない発話 $s_{u'k}$ を探索する。さらに $s_c(\text{start})$ と $s_{u'k}(\text{end})$ の差分絶対値 s_{\min} が閾値を以内であり、尚且つ u' が c の周辺ユーザ X_c に含まれている場合、 u' の発話に対して c が応答したという発話応答関係が成立したと解釈する。そして c から $u' = x$ への応答関係の総数を示す $z_x(\text{before})$ をインクリメントする。この処理によって c が他のユーザに対して行った応答の中で X_c 内の周辺ユーザそれぞれに対する応答のみに限定した応答数を得ることができる。9 行目から 13 行目では c の発話に対する各周辺ユーザの応答数を同様に計測している。最後に各周辺ユーザ x に対して、 c が x に応答した回数 $z_x(\text{before})$ と x が c の発話に応答した回数 $z_x(\text{after})$ の合計を c と x 間の会話応答頻度 z_{xc} として取得し、その集合 Z_c を c の会話応答頻度として取得する。21 行目で全てセンタユーザの会話応答頻度 Z を結果として返している。

このようなアルゴリズムによって各センタユーザの会話応答頻度を算出し、次節より説明するグルーピング処理の基礎データとして用いる。

3.4 会話応答頻度に基づくユーザグルーピング

これまで説明した高頻度発話者、同時発話者に関する情報と会話応答頻度に関する情報に基づいてユーザをクラスタリングし、単位時間内における会話グループの解析を行う。アルゴリズム 4 に Milvus のグルーピン

Algorithm 3 会話応答頻度の算出

Require: $U = \{u_1, u_2, \dots, u_i\}, C = \{c_1, c_2, \dots, c_{i'} | c \in U\}, \mathbf{X} = \{X_{c_1}, X_{c_2}, \dots, X_{c_{i'}} | X = \{x_1, x_2, \dots, x_a\}\}, \mathbf{S} = \{S_{u_1}, S_{u_2}, \dots, S_{u_i} | S = \{s_1, s_2, \dots, s_b\}\}$

Ensure: $\mathbf{Z} = \{Z_{c_1}, Z_{c_2}, \dots, Z_{c_{i'}} | Z = \{z_{x_1}, z_{x_2}, \dots, z_{x_a}\}\}$

function response_frequency($C, \mathbf{X}, \mathbf{S}$):

- 1: **for** c in C **do**
- 2: **for** s_c in S_c **do**
- 3: $s_{\min}(\text{before}) \leftarrow \min(|s_{u'k}(\text{end}) - s_c(\text{start})|)$
- 4: **comment:** $\{s_{u'k} \in S'_u \cap S'_u \in \mathbf{S}^{\bar{c}}, s_{u'k}(\text{end})$ is the timestamp when u' finished speaking at the k -th spoken, and $s_c(\text{start})$ equals the timestamp when c began speaking of $s_c\}$;
- 5: **if** $s_{\min}(\text{before}) \leq T_{\text{HERSHOLD}} \cap x \leftarrow u' \in X_c$ **then**
- 6: increments $z_x(\text{before})$, which is the count that c response to x 's spoken
- 7: **end if**
- 8:
- 9: $s_{\min}(\text{after}) \leftarrow \min(|s_c(\text{end}) - s_{u'k}(\text{start})|)$
- 10: **comment:** $\{s_{u'k} \in S'_u \cap S'_u \in \mathbf{S}^{\bar{c}}, s_c(\text{end})$ is the timestamp when c finished speaking of s_c , and $s_{u'k}(\text{start})$ equals the timestamp when u began speaking at the k -th spoken};
- 11: **if** $s_{\min}(\text{after}) \leq T_{\text{HERSHOLD}} \cap x \leftarrow u' \in X_c$ **then**
- 12: increments $z_x(\text{after})$, which is the count that x response to c 's spoken
- 13: **end if**
- 14: **end for**
- 15: **for** $z_{X_c}(\text{before})$ and $z_{X_c}(\text{after})$ **do**
- 16: $z_x \leftarrow z_x(\text{before}) + z_x(\text{after})$
- 17: add z_{x_c} to Z_c
- 18: **end for**
- 19: add Z_c to \mathbf{Z}
- 20: **end for**
- 21: **return** \mathbf{Z}
- 22: **end function**

グ処理の全容を示す。この処理には高頻度発話者の選定やユーザ毎の同時発話者の取得、会話応答頻度の算出処理も、これまで擬似コードとして示した関数を呼び出す形で含まれている。

grouping 関数の入力情報はユーザの集合 U とそのユーザの発話区間の集合 \mathbf{S} である。処理の結果として単位時間内における会話グループの集合 G が得られる。 G は会話グループ g の集合であり、 g は会話グループの参加者 $u' \in U$ を要素とする集合である。

grouping 関数では 2 行目から 9 行目に示されるように、まず高頻度発話者の集合 L 、同時発話者の定義情報 O 、該当単位時間内において発話を行わなかった無発話ユーザの集合 Q 、そして各センタユーザの会話応答頻度 \mathbf{Z} を取得する。各パラメータの取得は前節までに説明したアルゴリズムに基づいて行う。

これらのパラメータを利用して、ユーザのクラスタリングを行う。アルゴリズム 4 中 10 行目から 13 行目は

Algorithm 4 会話応答頻度に基づくユーザグルーピング

Require: $U = \{u_1, u_2, \dots, u_i\}$, $\mathbf{S} = \{S_{u_1}, S_{u_2}, \dots, S_{u_i} | S = \{s_k | s_k \text{ equals a set of timestamps that } u \text{ started speaking and finished speaking}\}$

Ensure: $\mathbf{G} = \{g_1, g_2, \dots, g_n | g \subseteq U\}$

- 1: **function** grouping(U, \mathbf{S}):
- 2: **for** u in U **do**
- 3: identify o_u who most frequently speaks overlaped with speak segments of u in the time unit
- 4: add o_u to O
- 5: **end for**
- 6: identify users Q that the set of a quiet user u_{quiet} ($S_{u_{\text{quiet}}} = \phi$)
- 7: get lead speakers L ($L \leftarrow \text{lead_speakers}(U, \mathbf{S})$)
- 8: define center users C and targets of each center \mathbf{X} ($\mathbf{X} \leftarrow \text{centers_and_targets}(U, L, O, Q)$)
- 9: calculate response frequency information \mathbf{Z} ($\mathbf{Z} \leftarrow \text{response_frequency}(C, \mathbf{X})$)
- 10: **for** m in $\bar{C} = \{m | m \in U \cap m \notin C \cap m \notin Q\}$ **do**
- 11: identify the center user c' based on \mathbf{Z} , that the speakings of c' are most frequently responded by m
- 12: add m to the previsional fixed group of c' , $g_{\text{pre}c'}$
- 13: **end for**
- 14: add each c to $g_{\text{pre}c}$ as the group member
- 15: identify the set of users U' that have not yet to belong any previsional fixed groups
- 16: **if** $U' \neq \phi \cap U \neq U'$ **then**
- 17: $G \leftarrow$ all previsional fixed groups $G' \cup \text{grouping}(U', \mathbf{Z})$
- 18: **else**
- 19: **for** u' in U' **do**
- 20: make new group g' only u' belongs
- 21: add g' to G
- 22: **end for**
- 23: **end if**
- 24: **return** G
- 25: **end function**

クラスタリング処理を示している。会話応答頻度に基づくユーザクラスタリング処理は基本的に、センタユーザではないユーザであり、尚且つ無発話ユーザでもないユーザ m と、 m との会話応答頻度の値が最大であるセンタユーザは互いに同じグループに参加するという基準でユーザを分類する。会話応答頻度が最大であるということは、他のセンタユーザと比べて最も発話に応答し合ったユーザである、ということを示している。本研究では会話応答頻度が高いユーザ同士は互いの発話に回答し合う機会が多い、すなわちそのユーザ同士は会話中であつたと解釈する。そのため、あるユーザが会話グループ毎に1人ずつ存在するセンタユーザと会話応答頻度が高いということは、そのセンタユーザが属する会話グループの参加者であると解釈できる。したがって本研究では、あるユーザが最大の会話応答頻度をセンタユーザ間に持つ場合、そのユーザはセンタユーザが

所属する会話グループに参加すると判断する。この基準に従いながら、ユーザを分類していき、会話グループのプロトタイプを生成する。

会話グループのプロトタイプを生成した時、しばしばプロトタイプグループ G' にも、無発話ユーザの集合 Q にも含まれないユーザが発生する。例えば、複数のセンタユーザが同時発話者として判別されており、それによってどのセンタユーザとも会話応答関係にないようなユーザが該当する。このようなユーザに対してもクラスタリング処理を施す必要があるため、既にクラスタリングが終了したユーザとは別にクラスタリングする。クラスタリングが完了していないユーザの集合 U' のみに対して、これまでの処理を再帰的に行う。このように再帰的にグルーピングを行うことによって、グルーピングから漏れてしまったユーザに対しても同様のグルーピング処理を通じて、所属グループを確定させることができる。

最終的に再帰的にグルーピングしたプロトタイプグループの集合 G' と無発話ユーザの集合 Q の論理和をグルーピング結果 G として取得する。無発話ユーザは誰とも会話を行っていないため、そのユーザしか参加者が存在しないグループを各自形成していると判別している。さらにこの時点で単位時間内に発話を行ったユーザは G' に定義されるプロトタイプグループのいずれかに所属している。この2つのグループの集合を合算したものを最終的なグルーピング結果とする。

本研究ではこれまでに説明したアルゴリズムでアドホック型会話のグルーピングを行う。

3.5 関連研究

本研究と同様の目的を持った関連研究及び既存手法についてまとめ、Milvus と比較考察する。

3.5.1 空間的基準によるグルーピング

人々が形成する集団を人々の位置情報や近接情報に基づいて空間的にグルーピングを行う手法は多くの既存研究で提案されている。この手法は人々が各自携帯する端末で定期的に取得した位置情報や各端末の相対距離情報を収集し、近接した人々同士を1つのグループであると判別する。以下に空間的な情報に基づいて人々のグループを検出する手法に関する研究を述べる。

The geography of taste

Francesco 氏らは人々が形成する大規模な群に焦点を当て、人々が所持する端末の位置情報に基づいて街中に発生した人々の群の検知を行なっている [17]。この研究は街中で何十人という人々が形成するような大規模な群の検知を対象としている。そのため、本研究が想定する会話グループのような小さな群を検知するためには検知粒度が大きく不十分である。

Find and Connect

Find and Connect[18] は、オフィス内で仕事からの従業員が所持する携帯端末の位置情報から得られる従業員同士の近傍状況からユーザグルーピングを行う。位置情報取得には携帯端末が観測した WiFi の電波強度とアクセスポイントの絶対位置情報を利用する測位方法を取り、屋内における各ユーザの近傍状況を割り出す。また、近傍中のユーザ同士をグループとして扱い、グループの参加者内のテキストメッセージシステムや、会議スペースの予約システム、といった社内サービスをグループ情報に基づいて提供している。

Find and Connect は小規模のユーザグループをリアルタイムに検知するという点で本研究の焦点と類似し

ている。しかし、ユーザの近傍情報に基づいたグルーピングはアドホック型会話において、会話への非参加者を無関係の会話グループにグルーピングしてしまうというように、実際の会話グループとグルーピング結果に乖離が生じる。したがって、Find and Connect のようなユーザの近傍情報に基づくグルーピング手法は本研究で定める機能要件を満たすことができない。

3.5.2 グルーピング専用端末を要する手法

グルーピングに利用する情報を汎用的な端末ではなく、専用のウェアラブル端末を用いて収集する手法は、手法が想定する特定環境下でユーザグルーピングを行う上で最も適した情報を得ることが可能である。以下に専用の情報収集端末をユーザが身に付けることでユーザグルーピングを行っている既存手法を挙げる。

Sociometer

Sociometer[1] は Face-to-Face の会話を検知し、会話コンテキスト情報を蓄積していくことで、実世界上でのソーシャルネットワークの構造化を目的としている。Sociometer は IR センサ、加速度センサ、小型マイクが内蔵したウェアラブル端末であり、装着者の発話状況や対面に位置する他のユーザを取得することができる。図 3.8 に Sociometer を装着時の光景を示す。

Sociometer はあくまで人々のコミュニケーションネットワークのモデル化を目的としているため、日常生活内で端末を実用化することよりも行動解析に最適な情報ソースの収集を重視している。そのため、Sociometer 自体は極めて日常生活内に浸透しにくい端末となり、日常生活内での常時装着はユーザにとって大きな負担になってしまう。

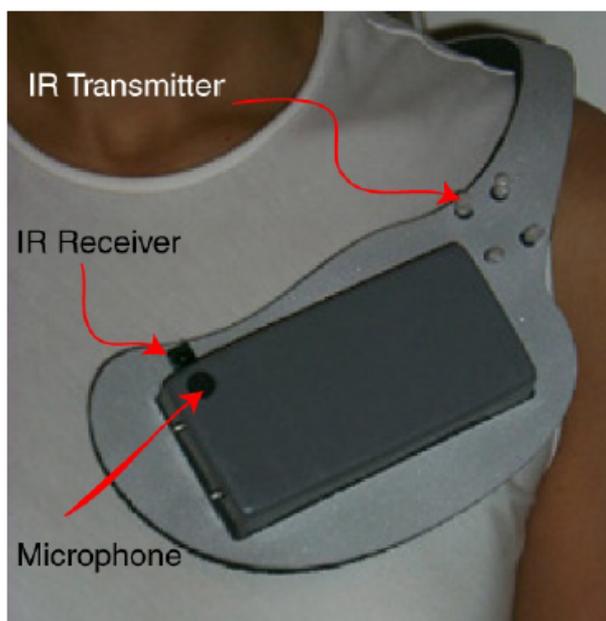


図 3.8 Sociometer 装着図 (引用:[1])

ビジネス顕微鏡

ビジネス顕微鏡 [19] は Sociometer と同様に、非汎用的な専用ウェアラブル端末を介して収集したセンサ情報に基づいて、オフィス労働者のコミュニケーションネットワークの構造化を図っている。ビジネス顕微鏡は IR センサや加速度センサを内蔵したカード状の端末を利用する。この端末をオフィスで働く各社員が首から提げるように装着した状態で常時収集したセンサ情報に基づき、社員同士の対面時間や動作を測定、蓄積する。これによって組織内の対面コミュニケーションのネットワークを定量的に分析することを可能にしている。

ビジネス顕微鏡は企業のオフィスといった限定的な環境下で、そこに属する人々のコミュニケーションネットワークの構造化、データ化に主眼を置いている。カード型の端末を使用し、装着者の負担軽減にも配慮しているが、専門的な端末を用いている以上、企業組織内というように端末の装着を強制できるような環境でないと導入することは難しい。

3.5.3 長期収集情報に基づく行動傾向解析

本研究と同様にユーザ周囲の情報の収集に一般的な携帯端末を利用した上で、日常生活内の会話全般においてその会話コンテキスト情報の検知を行う既存研究として、長期的に収集したセンサ情報を解析し、人々のソーシャルコミュニケーションの解析を試みている研究が挙げられる。この研究事例は長期的に収集したセンサ情報を一度に解析することにより、対象環境内で実際に行われているコミュニケーションの傾向をデータ化することを主眼に置いている。以下に研究の具体例を挙げる。

GroupUs

GroupUs[20] は、スマートフォンに搭載されている Bluetooth で周囲の端末間の電波強度を測定して近接センサとして利用した上で、長期に渡って記録したユーザ間の近接値情報に基づいてソーシャルコミュニケーションの解析及び可視化を行なっている。GroupUs ではオフィス内で従業員が行うコミュニケーションと自宅で家族とするコミュニケーションの両方を対象に、ユーザ間近接値情報に基づいて、各ユーザが誰とどのようなコミュニケーションを行う傾向にあるかを確率的に推論している。

日常生活内における会話を対象に、一般的な端末で収集した情報に基づいたユーザグルーピングを行なっているという点で、GroupUs は本研究と類似している。しかしながら、GroupUs はユーザ間近接値に基づいて空間的に近傍するユーザ同士をグループとして扱っている。そのため、会話コンテキスト情報の抽出手法として GroupUs が提案する手法を応用した場合、本研究が指摘するように会話グループに隣接する非参加者の誤検出が発生する。尚且つ、GroupUs は長期的に収集した情報に基づいた解析に焦点を当てているため、本研究が想定するアプリケーションで利用可能なコンテキスト情報を提供することはできない。

3.5.4 音声解析による手法

これまで例示した既存研究と異なり、実際の会話音声解析することで日常生活内における会話の検出と会話相手の推測を試みる研究も本研究と関連する研究として挙げられる。音声解析結果に基づく既存手法はユーザが所持する端末の内蔵マイクで取得した音声を基礎データとして用いており、それを波形解析した後、独自の判別式に当てはめることで会話検知を行う。以下にその具体的な研究事例を示す。

端末音声の相互相関に基づくアドホック会話の検出

岡本氏らによる研究 [21] はユーザが携帯するセンシング端末としてラップトップ PC を利用し、その内蔵マイクで取得した音声情報を基にユーザ同士の会話検知と会話参加者の判別を行う。この研究ではユーザ周囲の音声情報を各ユーザの端末で取得し、取得音声情報間の相互相関係数を求める。この相互相関係数が高い音声記録した端末同士を互いに近傍端末とし、その端末の所有者を互いに会話への参加者であると判別する。

この研究で提案されている手法は一見、日常生活内のアドホック型会話のグルーピングに成功しているように見える。しかし、この手法はマイクを近接センサとして利用しているだけであり、他の既存手法と同様に近傍の非参加者を誤検知する恐れが非常に高い。この件に関しては、筆者らも [21] 内で言及している。したがって、本研究の定める機能要件を十分に満たすことができない。

DarwinPhones

DarwinPhones[2] は、スマートフォン上でセンサ情報の収集と収集した情報に基づいたコンテキスト情報解析を高精度に行うためのフレームワークである。図 3.9 に DarwinPhones が提案する基本概念を図示したものを示す。DarwinPhones はコンテキスト解析処理において、1) 各スマートフォンが保持しているコンテキスト解析のための学習モデルを近傍のスマートフォン同士で相互共有した上で、2) それぞれが取得したセンサ情報に基づいてコンテキスト情報の推論処理を行い、3) 推論結果を再度近傍の端末間で相互共有して精度補正を行う、という 3 段階の処理構造を設けることで、コンテキスト情報解析の精度を飛躍的に高めるといってコンテキスト情報解析系の処理モデルである。DarwinPhones では、その被検対象として日常生活環境内のアドホック型会話における話者認識処理に着目し、既存の話者認識アルゴリズムに DarwinPhones フレームワークを組み込んだ話者認識システムを構築した上で精度実験を行い、高い認識精度を記録している。

DarwinPhones は被験対象としてアドホック型会話の話者認識に焦点を当てているが、筆者らが論文中で述べているように、DarwinPhones は話者認識及び会話コンテキスト情報の抽出を目的とする研究ではなく、あくまでコンテキスト解析処理全般におけるフレームワークの提案に主眼を置いている。また、DarwinPhones が被験対象として着手しているのはあくまで話者認識アルゴリズムのみであり、本研究が目的とする会話コンテキストの抽出やアドホック型会話における参加者グルーピングを対象としているわけではない。本研究で提案する手法の一部に DarwinPhones の実装成果が共通する部分も見られるが、本研究とは研究目的が明確に異なっている。

3.6 本章のまとめ

本章ではアドホック型会話のグルーピングにおいて、本研究が提案するグルーピングアルゴリズムを説明した。本研究が提案するグルーピングは各ユーザの発話区間を時系列的に比較し、発話の開始時刻と終了時刻の関係から、誰が誰に何度応答を行ったか、という会話応答頻度に基づいて行う。この時会話応答頻度が適切に算出できるよう、単位時間における発話量の多い高頻度発話者と、発話区間が重複する同時発話者の情報から会話応答頻度の算出範囲を限定する。これによって実際の会話応答関係との食い違いを低減して、会話応答頻度の算出を行う。会話応答頻度に基づいたユーザクラスタリングでは会話応答頻度が高いユーザ同士をグループとして分類する。さらにクラスタリングに漏れたユーザに対して再帰的にグルーピング処理を行う。

本研究の提案するアルゴリズムは既存の提案手法と異なり、日常的な端末を利用して収集可能な情報に基づいてグルーピングを行うことができる。尚且つ会話の応答関係に基づいてグルーピングを行うことで、ユーザ

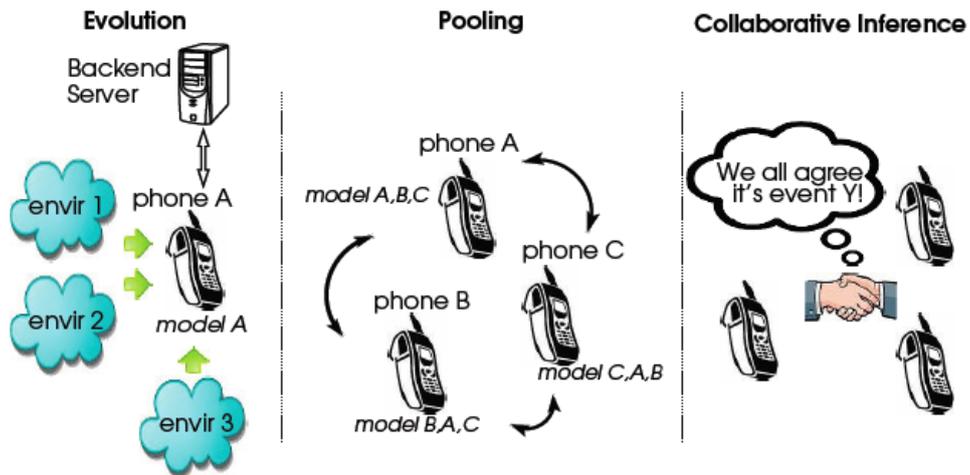


図 3.9 DarwinPhones フレームワークの基本概念 (引用:[2])

の位置情報や近接情報に基づいてグルーピングする既存手法では対処困難な、会話グループに近傍する非参加者を排除しながらのグルーピングを可能にすることができる。

次章では本研究が提案したグルーピング手法に基づいたユーザグルーピングシステムの設計と実装について説明する。

第 4 章

設計・実装

3 章で説明したグルーピング機構 Milvus に基づき、会話グループのリアルタイムグルーピングを実現するシステムの構築を行った。本章では本グルーピングシステムの設計と実装について説明する。

4.1 設計

本グルーピングシステムのハードウェア構成とソフトウェア構成について説明する。

4.1.1 ハードウェア構成

グルーピングシステムのハードウェア構成について、その構成図を図 4.1 に示した上で説明する。本グルーピングシステムでは、ユーザが携帯する端末となるスマートフォンが音声情報と位置情報の取得を常時行う。取得した情報は 3G 回線、もしくは WiFi 回線と有線回線網を通じて、遠隔のグルーピングサーバへと定期的に送信される。グルーピングサーバでは各ユーザ端末から送信された情報に基づいてグルーピング処理を行い、その結果はサーバ内部の、もしくは有線で接続された会話グループ情報データベースへと保存される。さらにこのグルーピングサーバはグループ情報配信サーバも兼ねており、ユーザ端末からの要求に従って、データベース内のグループ情報を送信する。なお、ユーザ端末がグループ情報を取得した後の端末上での処理はアプリケーションに依存する。

4.1.2 ソフトウェア構成

上述したハードウェア構成上に構築されるグルーピングシステムにおいて、センサ情報の取得送信及びサーバ上でのグルーピング処理を行う各ソフトウェアモジュールとその構成を図 4.2 に示す。入力情報はユーザ端末がマイクから取得したユーザの会話音声、端末内の位置測位機器から得られた端末位置情報、そして本グルーピングシステムが定義した、ユーザと端末を一意に関連付けるユニークなユーザ ID である。これらの情報を端末からグルーピングサーバへとネットワークを介して定期送信する。

グルーピングサーバソフトウェアを構成するモジュールは、大別すると、物理近接ユーザ判定部、センタユーザ・周辺ユーザ定義生成部、会話応答頻度算出部、会話グループ判定部の 4 つのモジュールであり、各モジュールを複合した機構が Milvus である。サーバ上で処理されるグルーピングプロセスは、物理近接ユーザ判定部において各ユーザ端末から送信された位置情報に基づき、端末の位置情報が近接しているユーザの集団を得る。この近接ユーザの集団内で同じくユーザ端末から送信された発話区間に基づいたグルーピング処理を



図 4.1 ハードウェア構成図

他の3つのモジュールで行う。センタユーザ・周辺ユーザ定義生成部は3章で詳述した会話応答頻度算出のためのセンタユーザ及び周辺ユーザ群の選別を行う。このモジュールで生成された情報は会話応答頻度算出部に渡される。会話応答頻度算出部はセンタユーザ、周辺ユーザの定義情報に基づき会話応答頻度を算出した上で、会話グループ判定部に送信する。会話応答頻度情報を渡された会話グループ判定部は提案手法のグルーピングに基づき、会話グループの判定を行う。最終的に、解析されたグループ情報は会話グループ情報のデータベースストレージに保存される。これらのグルーピング処理はイベントドリブンモデルによる実行ではなく、定期実行されることを前提に設計されている。

以下で各モジュールに関して詳しく説明する。

4.1.3 クライアントソフトウェア

スマートフォン内で動作するクライアントソフトウェアは単位時間における発話区間情報の生成と端末位置情報の取得、及びセンサ情報の定期送信を行う機能が動作する。各機能はスマートフォン上でバックグラウンドサービスとして動作する。発話区間情報生成部はマイクから常時入力される音声情報から、端末を携帯するユーザが発話を行った区間の開始時刻と終了時刻を記録する。位置情報取得は端末が搭載するGPSや無線電波強度に基づく位置測位技術を使用して定期的に端末の緯度経度を取得する。更に、ユーザアカウントIDをソフトウェア保存領域から取得する。これらの情報をネットワークでサーバソフトウェアの仕様にしたがってクエリ文字列としてエンコーディングし、ネットワークを通じて送信する。

これらのバックグラウンドサービスの他に、各種サービスの起動停止用コントロールインターフェイスを提供するフロントアプリケーションを備えている。

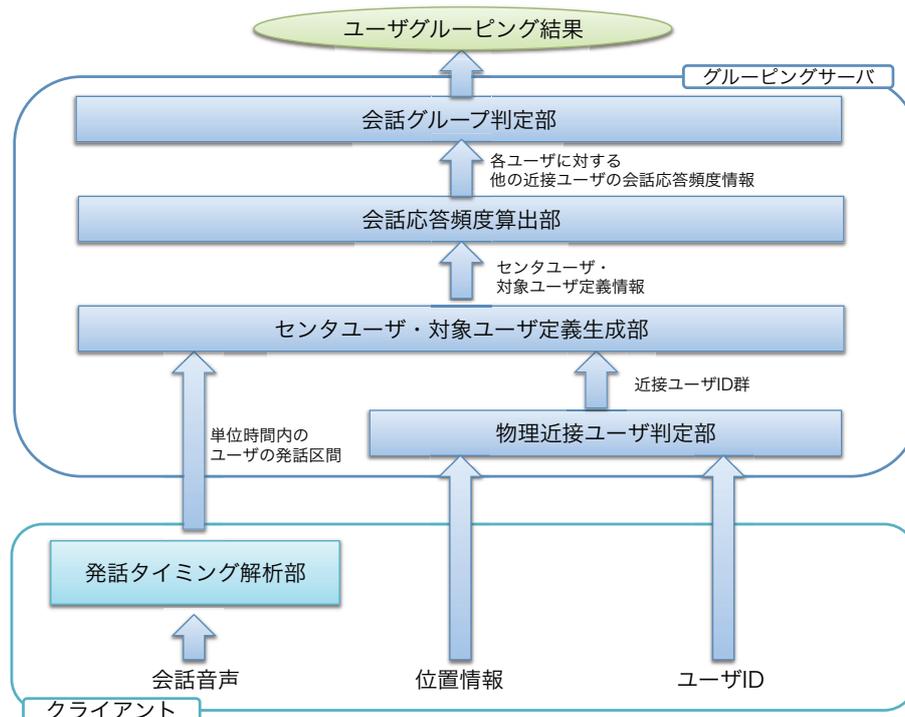


図 4.2 会話グルーピング処理に関するソフトウェア構成図

4.1.4 センタユーザ・周辺ユーザ定義生成部

センタユーザ・周辺ユーザ定義生成部では近傍ユーザ内で、単位時間におけるセンタユーザの選出とセンタユーザ毎の周辺ユーザの定義を行う。選出処理は前章で説明したアルゴリズムに則って行われる。そのため、本モジュールでは高頻度発話者の選出、同時発話者の定義情報といった、定義生成に必要なパラメータ情報の生成も行う。処理の結果はセンタユーザ ID と周辺ユーザ ID のリストの key-value リストとして表現され、各ユーザの発話区間情報と共に会話応答頻度算出部へと渡される。

4.1.5 会話応答頻度算出部

会話応答頻度算出部は入力として与えられたユーザの発話区間情報とセンタユーザ、周辺ユーザの定義情報に基づいて、単位時間におけるセンタユーザを中心とした会話応答頻度の算出を行う。会話応答頻度の算出アルゴリズムは3章で詳述した手順で行う。これによって得られる会話応答頻度情報をセンタユーザ、周辺ユーザに関する情報と共に会話グループ判定部へと送信する。

4.1.6 会話グループ判定部

会話グループ判定部は会話応答頻度算出部の処理によって得られる、各センタユーザを中心とした周辺ユーザとの会話応答頻度情報に基づいて、会話グループの参加者をクラスタリングするモジュールである。クラス

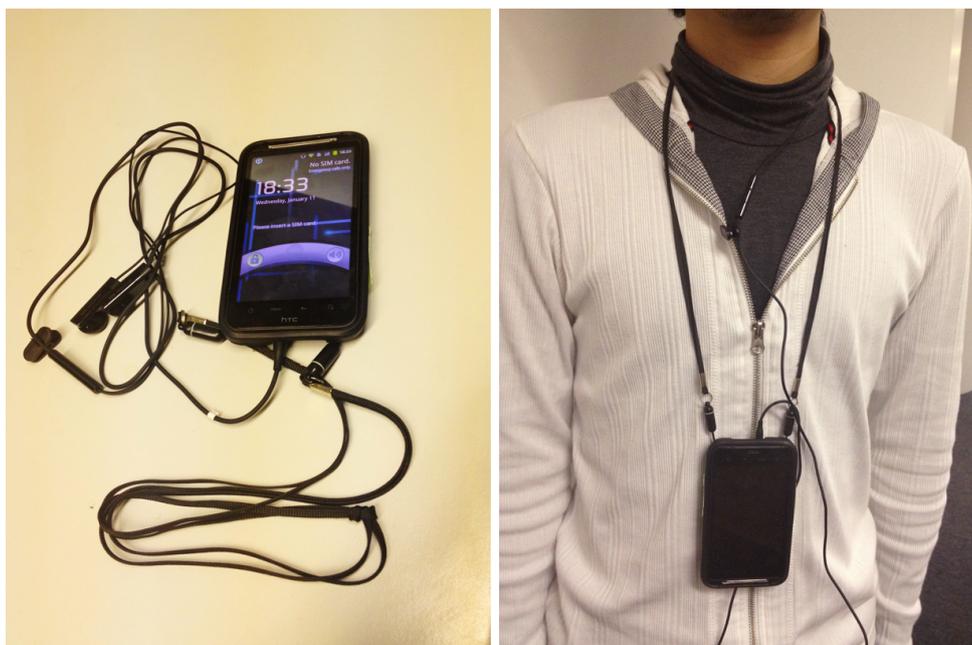
タリング処理は3章で述べた手順で行い、クラスタリング対象ユーザが参加する会話グループが全て判別できると、得られたグループ情報を保存領域へと保存する。クラスタリング処理が完了しても一部のユーザの会話グループが判別しきれない場合は、残りのユーザ情報をセンタユーザ・周辺ユーザ定義生成部へと送信し、再帰的にグルーピング処理を行う。最終的にグループ情報が収束した場合に対象ユーザに関するグルーピング処理が完了したとして処理を終了する。

4.2 実装

上述した設計に基づき本グルーピングシステムの実装について説明する。ハードウェアにおいては実装に使用した機器構成を挙げ、ソフトウェアについては各モジュールの実装について述べる。

4.2.1 実装環境

本システムの実装では各ユーザが携帯するセンシング端末とグルーピングサーバが動作するサーバ機器の2つの機器を使用した。センシング端末には HTC コーポレーションの Desire HD[22] を用いた。このスマートフォンに付属する通話用イヤホンマイクを本体のジャックに装着して音声取得に利用した。更に図 4.2.1 のように本体にストラップを取り付けユーザが身に付けやすい状態にして使用した。ただし、端末を首から提げて装着する事が前提ではなく、端末本体に関してはユーザの衣服のポケットに入れるといった携帯方法も可能である。使用端末の環境を表 4.1 に示す。



(a) イヤホンマイクを装着した端末

(b) 端末装着例：端末を首から提げた時の光景

図 4.3 使用端末

本端末は実験用ファームウェア CyanogenMod7.1[23] を動作させた状態で利用した。CyanogenMod7.1 は Desire HD のファームウェアの基礎となる Android OS version 2.3.4[24] に基づいて実装されたオープンソー

スファームウェアである。Desire HD に標準搭載されるファームウェアが商用にカスタマイズされているのに対して、このファームウェアは余分な機能が排除されていることによりソフトウェアの実装が容易であるため採用した。クライアントソフトウェアは Android version 2.3.4 準拠の Dalvik VM 上で動作する。さらに、ソフトウェアの実装には Java SE Development Kit 6.0 と Android SDK 2.3.4 を使用した。

使用端末	HTC Desire HD
ファームウェア	CyanogenMod7.1
言語	J2SDK 1.6.0
実行環境 VM	DalvikVM
SDK	Android SDK 2.3.4

表 4.1 センシング端末実装環境

グルーピングサーバシステムが動作するサーバ機器は MacBookAir を使用した。表 4.2 にサーバ機器環境を示す。ソフトウェア実装言語は Python version 2.7.2 (GCC 4.2.1 Apple Inc. build 5666 dot 3 on darwin) を使用し、同言語の標準ライブラリを利用した。

使用機器	MacBookAir
プロセッサ	1.8 GHz Intel Core i7
主メモリ	4 GB 1333 MHz DDR3
OS	Mac OS X Lion 10.7.2
開発言語及び実行環境	Python version 2.7.2

表 4.2 サーバシステム実装環境

4.2.2 クライアントソフトウェア

センシング端末内で動作するクライアントソフトウェアは、Android アプリケーションとして実装される。音声取得及び発話区間解析を行うモジュール、端末位置情報を取得するモジュールは Android SDK で提供される android.app.Service クラスを継承したクラスとして実装される。この実装によってセンシング処理をバックグラウンドで動作させることができ、ユーザは自由に他の端末アプリケーションを利用可能となる。

発話区間解析部

発話区間解析部ではマイクから入力された音声情報をファームウェア内部で離散数値データ化された情報を基に、端末ユーザが発話中であるか判定する。発話判定結果は単位時間中一時的にソフトウェア内に保存され、単位時間が経過した際に発話区間情報に整形される。最終的に、グルーピングサーバへセンシング情報を送信するモジュールに発話区間情報を渡す。本システムでは外部の集音マイクを介して音声を取得することから、入力された音声は全て端末ユーザの発話によるものであるとする。

本論文の対象領域は会話参加者のグルーピングであるため、話区間判定用の音声解析には話者認識手法として提案されている既存アルゴリズムを利用する。本システムは話者認識手法として、比較的演算コストが低く実装が容易な音声波形振幅と零交差数に基づく音声区間検出法 [13][14] を利用する。この手法は入力音声の振

```

1 public boolean isVoiceActivity(double[] data) {
2     boolean v = false;
3     for (int i = 0; i < data.length; i += this.chunkSize) {
4         int zeroCrossCount = 0;
5         for (int j = 0; j < this.chunkSize - 1; ++j) {
6             double value = data[j + i];
7             double next = data[j + i + 1];
8             if ((Math.abs(value) >= this.amplitudeThreshold) && ((value*next) < 0)) {
9                 ++zeroCrossCount;
10            }
11        }
12    }
13    if (zeroCrossCount >= this.zeroCrossThreshold) {
14        v = true;
15    } else {
16        v = false;
17    }
18    return v;
19 }

```

図 4.4 音声区間検出関数

```

1 public void onReady(List<Boolean> voiceAcitivityList, double step) {
2     double period = 0.0;
3     double start = -1.0;
4     double end = -1.0;
5     ArrayList<double[]> speechSections = new ArrayList<double[]>();
6     for (Iterator<Boolean> i = voiceAcitivityList.iterator(); i.hasNext();) {
7         boolean b = i.next();
8         if ((start < 0.0) && b) {
9             start = period;
10        }
11        else if ((start >= 0.0) && (end < 0.0) && !b){
12            end = period;
13            double[] section = {start,end};
14            speechSections.add(section);
15            start = -1.0;
16            end = -1.0;
17        }
18        period += step;
19    }
20    ....
21 }

```

図 4.5 発話区間情報の生成

幅の大きさと波形グラフが異符合に遷移する回数（零交差数）に基づいた音声入力判定方法である。図 4.4 に音声入力判定関数の実装コードを示す。実装コードは固定長配列に格納された離散波形データを引数とする。この配列はマイクから入力された音声情報の 1 フレーム分の離散データであり、各値が音声波形の振幅を表している。この配列内の値を順に評価し、1 フレームの零交差数を計数する。各値の評価は 8 行目から始まる条件式で行われる。この条件式は配列内の任意の要素に対して、振幅の大きさが閾値を超えていて、且つ次の要素との乗数が 0 よりも小さい、すなわち零交差をしているか判定する。そして判定結果が真である場合が閾値を超えたフレームを発話区間として判定する。

この関数によって単位時間内に得られた音声情報に対して 1 フレーム毎に音声入力判定を行うと、フレーム内の音声入力の有無を示す 2 値リストが得られる。この 2 値リストをパースすることにより、単位時間内における発話区間を取得することができる。図 4.5 に 2 値リストを発話区間情報にパースする実コードを示す。

コード中の引数である変数 `voiceAcitivityList` は音声入力の 2 値リストであり、変数 `step` はサンプルレートと離散データ数から得られる音声 1 フレームあたりの秒数である。この関数では 2 値リスト内で真が連なっ

```

1  {'data': '[14.25,14.375],[15.125,15.25],[15.375,15.625],[15.75,15.875],[16.625,16.75],
2  [18.625,18.875]'},'email':'charlie.htsfc@gmail.com','latitude':35.3881125,
3  'longitude':139.4253995,'name':'charlie.htsfc','time':'2011-12-20 09:34:04.676773'}

```

図 4.6 センサ情報更新クエリ文字列

```

1  {'name':'charlie.htsfc@gmail.com'}

```

図 4.7 グループ情報取得クエリ

ている部分だけを抜き出し、抜き出した配列の先頭と末尾のフレームに対してリスト先頭から数えた時の経過秒数を記録する。その結果を可変長配列 `speechSections` に格納する。最終的に `speechSections` には、音声入力区間の開始及び終了時の相対時刻で構成される 2 要素の配列の集合が格納される。この格納された配列 1 つが単位時間内におけるユーザの発話区間を表している。この処理によって得られる発話区間情報をセンサ情報更新クエリ文字列に組み込み、グルーピングサーバへ送信する。

ユーザ情報

本システムではユーザとセンシング端末を関連付けるユーザ ID として google アカウントを利用した。Android は標準アプリケーションを利用するためにプライマリの google アカウントを登録することを要求する。そのため、Android を搭載したスマートフォンは 1 つの google アカウントと端末が関連付けられている。本システムでは端末とユーザを一意に決定できる識別子としてこの google アカウントを利用した。google アカウントは `google.com` ドメインの電子メールアドレスである。この電子メールアドレスをユーザ ID とした。

センサ情報更新クエリの生成

上述した発話区間情報、ユーザ情報といった情報をグルーピングサーバへと送信する。本グルーピングシステムでは情報送受信処理の簡略化のため、軽量データ交換フォーマットである JavaScript Object Notation (JSON) でエンコードしたクエリ文字列を送信することによってセンサ情報送信を行う。本グルーピングシステムにおけるセンサ情報更新クエリの例を図 4.6 に示す。クエリ中 `data` キーの値となる 2 重配列文字列が発話区間情報を表しており、重配列内の各配列が発話区間の開始終了時刻を示す。発話区間の値は `time` キーで示されるタイムスタンプからの経過秒数である。すなわち図 4.6 で示される発話区間の先頭は同クエリのタイムスタンプから 14.25 秒経過後にユーザが発話を開始し、14.375 秒経過後に終了したことを表している。クライアントソフトウェアはこのクエリを定期的にサーバへ送信する。

グループ情報の取得

サーバソフトウェアはグルーピング処理を行うと共にアプリケーションに対してグルーピング結果として得られるグループ情報を取得するインターフェイスを提供する。本システムではアプリケーションがネットワークを介して JSON 形式のクエリ文字列を送信することでグループ情報を要求できるグループ情報提供インターフェイスを提供する。具体的には、アプリケーションからユーザ ID が含まれた図 4.7 といったクエリを送信することによって、そのユーザが参加しているグループの情報を取得することができる。

サーバソフトウェアからアプリケーションに返却されるグループ情報は図 4.8 に例に挙げる JSON 文字列で表現される。この文字列はアプリケーションが送信したユーザ ID が参加するグループの参加ユーザの ID

```

1  {'group': ['alice.htsfc@gmail.com', 'charlie.htsfc@gmail.com'],
2  'daisy.htsfc@gmail.com'], 'latitude': 35.3877841, 'longitude': 139.4253565,
3  'time': '2011-12-20 19:21:02.272777'}

```

図 4.8 グルーピング結果

のリストを表す。アプリケーションはこの文字列をパースすることによって、グループ内の他のユーザ ID を得ることができる。

4.2.3 Milvus

本グルーピングシステムのサーバソフトウェアの大部分は、Milvus を構築するモジュールである。以下に各モジュールの実装を説明する。

物理近接ユーザ判定部

物理近接ユーザ判定部はセンシング端末から送信されたクエリに含まれる位置情報に基づき、端末位置に近いユーザの集団を抽出する。本モジュールは端末の緯度経度からユーザ間の距離を算出し、その距離が近いユーザ同士を近傍するユーザとして判別する。本システムにおける近傍判定はまず任意のユーザと他のユーザの距離を順に算出する。そして各ユーザを中心として、距離が閾値を下回るユーザの和集合を近傍するユーザの集合として取得する。本システムにおける距離算出式は数式 4.1 の Hubeny の公式 [25] に基づいて行った。数式中の y , x は緯度経度、 M は子午線曲率半径、 N は卯酉線曲率半径を表し、 M , N は WGS84 定数に基づき算出した。

$$D = \sqrt{\{(y_1 - y_2)M\}^2 + \{(x_1 - x_2)N \cos \frac{y_1 + y_2}{2}\}^2} \quad (4.1)$$

センタユーザ・周辺ユーザ定義生成部

センタユーザ・周辺ユーザ定義生成部は各端末から送信された発話区間情報から高頻度発話者と各ユーザの同時発話者を特定した上で、センタユーザと周辺ユーザの選別処理を行う。処理結果はセンタユーザのユーザ ID を key とし、センタユーザに対する周辺ユーザは、そのユーザ ID の配列を value とする辞書オブジェクトとして表現される。この辞書オブジェクトをセンタユーザ及び周辺ユーザの定義情報とし、会話応答算出部へ各ユーザの発話区間情報と共に送信する。

会話応答頻度算出部

会話応答頻度算出部はセンタユーザ、周辺ユーザを定義する辞書オブジェクトを定義生成部から受け取り、その定義情報に従ってセンタユーザを軸とする会話応答頻度算出処理を行う。算出アルゴリズムは 3 章で詳述したアルゴリズムに基づく。会話応答頻度情報は各センタユーザのユーザ ID を key とする辞書オブジェクトとして表現される。このオブジェクトは value に周辺ユーザのユーザ ID を key とし、そのユーザの会話応答頻度を value とする辞書オブジェクトを持つ、2 重辞書オブジェクトである。このオブジェクトを会話グループ判定部に送信し、会話グループの判定に利用する。

会話グループ判定部

会話グループ判定部は会話応答頻度情報に基づき、会話グループの判定を行う。グルーピングアルゴリズムは3章で詳述したアルゴリズムに基づく。判定結果として得られるグループ情報はグループ参加者のユーザIDの配列で表現される。この情報を会話グループ情報として抽出する。

4.3 本章のまとめ

本章は会話応答頻度に基づく会話グルーピング機構である Milvus に基づくグルーピングシステムの設計と実装の詳細を説明した。次章では本グルーピングシステムを用いた、会話グループ情報に基づくコンテキストウェアシステムの例について述べる。

第 5 章

会話コンテキスト情報の利用

前章にて説明したグルーピングシステムと連携し、会話コンテキストに基づいてサービス提供を行うアプリケーションの試作を行った。本章では 2 章で述べたアプリケーションシナリオと照らし合わせ、会話コンテキストの利用対象についてまとめる。さらに試作を行ったアプリケーションの動作について説明する。

5.1 会話コンテキストの利用

まず、Milvus による会話コンテキストのコンテキストアウェアシステムにおける応用についてまとめる。本研究において例示する応用方法は 3 つである。コンテキストアウェアシステムが提供するサービスの提供対象指定への応用、仮想空間内の情報へのアノテーションへの利用、そして実世界上で行われる対人コミュニケーションの情報化での利用である。

5.1.1 サービス対象の指定

昨今のコンテキストアウェアシステムはユーザ単体での利用を想定するシステムだけでなく、複数ユーザの集団に対してサービスを提供するシステムが提案されている。例えばその場に集まったユーザのグループに対して、各ユーザの趣向に基づいた飲食店の情報をレコメンドするグループレコメンドサービスや会社内の共有スペースで行われるミーティングの議事録を参加者の音声情報に基づいて作成支援するグループウェア、対面で会話をしている相手との即興的なファイル共有環境を提供するクラウドサービス等が考えられる。

これらのようなユーザのグループに対してサービスを提供するコンテキストアウェアサービスでは、ユーザへのサービス提供範囲を判定する必要がある。すなわち、人々の中からサービスを要求するユーザを判別し、どのユーザに対してサービスを提供するかシステムが判別する必要があるということである。例えば会話相手間での即興的ファイル共有環境を提供するサービスでは、誰と誰が会話中であるか判別し、ファイルの共有を行うユーザを絞り込むことが必要となる。

これに対して、コンテキストアウェアシステムがサービス提供を行うユーザは、会話を行いながらシステムを利用するというようにユーザ間でコミュニケーションを取りながらシステムを利用することがしばしば見受けられる。カメラを趣味とする人々が懇談をしながら撮影した写真を会話相手に共有する場面や、偶然集まった友人と会話をしながらレコメンドサービスを利用して飲食店を探索する場面といった例が挙げられる。

このようなシステムでは会話中のユーザのグループに対してサービス提供を行うため、会話中のユーザ同士をグループとしてシステムが判別出来る必要がある。これについて、Milvus による会話グルーピングを利用

し、コンテキストウェアサービスがグルーピング結果に基づいてサービスを提供することで、システムは自律的にサービスを提供することが可能となる。

さらには、音声認識技術やウェアラブルコンピューティング技術と複合することによって、より自然な形でユーザがサービスの提供対象を指定できると期待できる。例えば情報共有システムでは、2章で説明したアプリケーションのように、情報の送信対象を指定する際に“皆に”情報を送信とか“田中君に”画像を送って、といった曖昧な指示語を用いた自然な音声操作によるシステムのインタフェースを実現できると考えられる。このような曖昧な指示語を含む音声操作であっても、情報に基づいてユーザが意図する指定対象を推測することで、サービス対象の指定を可能とすることができる。これにより従来のシステムのように煩わしい操作をユーザに要求する必要がなくなり、ユーザはより自然な操作でサービスを利用可能となる。

このように、会話コンテキストはコンテキストウェアシステムにおいて、ユーザへのサービス提供範囲を指定する情報として役立つ。

5.1.2 仮想情報の注釈

情報検索システムにおいて、検索の利便性向上のため、検索対象となる情報にタイムスタンプやユーザ情報といった注釈を付け、検索クエリとの照合に用いる方法が採用されている。近年では位置情報やセンサ情報を仮想情報の注釈として用いることにより、実世界上のコンテキストに基づく情報取得やより高次元な検索クエリの利用が可能になりつつある。

この仮想情報の注釈として会話コンテキストを用いることにより、実世界上のコミュニケーションに関するコンテキストに基づいた情報検索が可能になると期待できる。ウェブ上の記事やコンテンツ、ユーザのパーソナル情報、コンピュータ内の文書ファイル等に会話コンテキストの注釈を施すことによって、会話が起こった場所や時刻、会話相手のユーザ情報に基づいてこれらの情報をコンピュータから検索できる。例えば、学会などで催される懇親会において懇談を行った相手と連絡先が記されたパーソナル情報を交換するような場面で、交換した情報に会話コンテキストを関連付けるといった利用方法が考えられる。ユーザが後日に懇親をした相手全員に挨拶のメールを送信する時、会話コンテキストに基づいて検索を行うことで、懇親会にて会話を行ったときに交換された連絡先を従来よりユーザの経験則に基づいた検索クエリによって見つけ出すことができる。従来のタイムスタンプや低次元な位置情報に基づいて情報を検索する場合と比較して、ユーザはより想起しやすい情報を利用して検索することができる。さらに、会話という実世界で起こった具体的なイベントによる検索であるため、懇親会の行われた時間帯と会話コンテキストのタイムスタンプを比較しながらユーザが情報を探索するような煩わしい作業を回避する事ができる。

このように、会話コンテキストを情報検索システムの注釈として利用することによって、従来よりも実世界のコンテキストに則したシステムの実現が期待できる。

5.1.3 日常生活内のソーシャルネットワーク情報の蓄積

実世界で人々が行うコミュニケーションに関して情報取得を行い、コミュニケーションを情報化するという試みが行われている [19][20]。これらの研究では長期的に収集した会話コンテキストに基づいて人々のコミュニケーションの傾向を解析することや、オフィス内のコミュニケーションに焦点を当てるなどしてコミュニケーションを行うユーザのモニタリングやマネジメントに利用することを目的としている。

このような実世界上のソーシャルネットワークに関する情報の解析に、本研究が提案する日常生活内の会話

グループ情報を利用することにより、これらの研究とは違った利用法を想定可能な解析システムを実現できる。その一例として日常生活内の情報をユーザのパーソナルクラウドストレージに蓄積するライフログの構築が挙げられる。これは本研究が提案するシステムによる会話グルーピングが日常生活内の会話に注目している点と、ユーザと解析結果にリアルタイム性が確保される設計を行なっている点による。

これによって、ユーザは自身に自由にアクセス可能な、自分を取り巻くソーシャルネットワークの情報のログシステムを得ることができる。このログシステムの会話コンテキストと従来のライフログを基板としたサービスが複合されることにより、過去の会話コンテキストに基づくコンシューマ向けサービスやエンターテインメントコンピューティング分野での利用が期待できる。

5.2 会話コンテキストに基づくコンテンツ共有システムの試作

上述したように、会話コンテキストはコンテキストアウェアシステムをはじめとする様々なサービスへの応用が期待される。そこで本研究では、提案手法によるグルーピングシステムを利用し、会話コンテキストに基づくアプリケーションを試作した。試作したアプリケーションは、2章でアプリケーションシナリオに挙げた、会話グループ内における情報共有アプリケーションである。本節では、試作アプリケーションについて詳述する。

5.2.1 試作アプリケーションの動作

試作アプリケーションは、スマートフォンを介して会話中の相手に画像ファイルを一齐共有するアプリケーションである。ユーザは自分のスマートフォンに外部マイクを取り付け、音声情報を常時取得可能な状態で携帯してアプリケーションを利用する。アプリケーションは図 5.2(a) に示す、共有対象の画像ファイル選択のためのインタフェースと画像ファイルを会話グループの参加者が持つ端末へと送信する機能を提供する。

アプリケーションのバックグラウンドでは、本研究が提案するグルーピングシステムのクライアントシステムが動作しており、ユーザの音声情報、位置情報、ユーザ ID から成るセンサ情報を定期的にサーバへ送信する。また、試作アプリケーションにはグルーピングサーバと通信して、ユーザのグループ情報を取得する機構が組み込まれている。これにより、アプリケーションからグルーピングサーバへグループ情報取得クエリを送信することができる。

グルーピングサーバとの通信により得られるグループ情報に基づき、試作アプリケーションは画像情報の送信を行う。試作アプリケーションによる画像情報の共有は、ユーザが画像選択インタフェースを介して共有画像を選択することによって共有処理が開始される。ユーザによって画像が選択されると、アプリケーションはグルーピングサーバへユーザのグループ情報を問い合わせる。この通信によって得られるグループ情報からユーザと同じ会話グループに所属する人々の ID を割り出し、その ID に基づいて画像情報を送信する。これによって、ユーザは会話中の相手に画像情報を共有する事ができる。

5.2.2 アプリケーションの実装

試作アプリケーションの構成図を図 5.1 に示し、表 5.1 及び 5.2 に実装環境を示す。試作アプリケーションは Android 端末内で動作するスマートフォンアプリとして実装されており、動作端末はグルーピングシステムへ送信するセンサ情報の収集と同時に、グループ情報に基づいたコンテキストアウェアシステムのサービス提供端末としても使用することが可能となる。

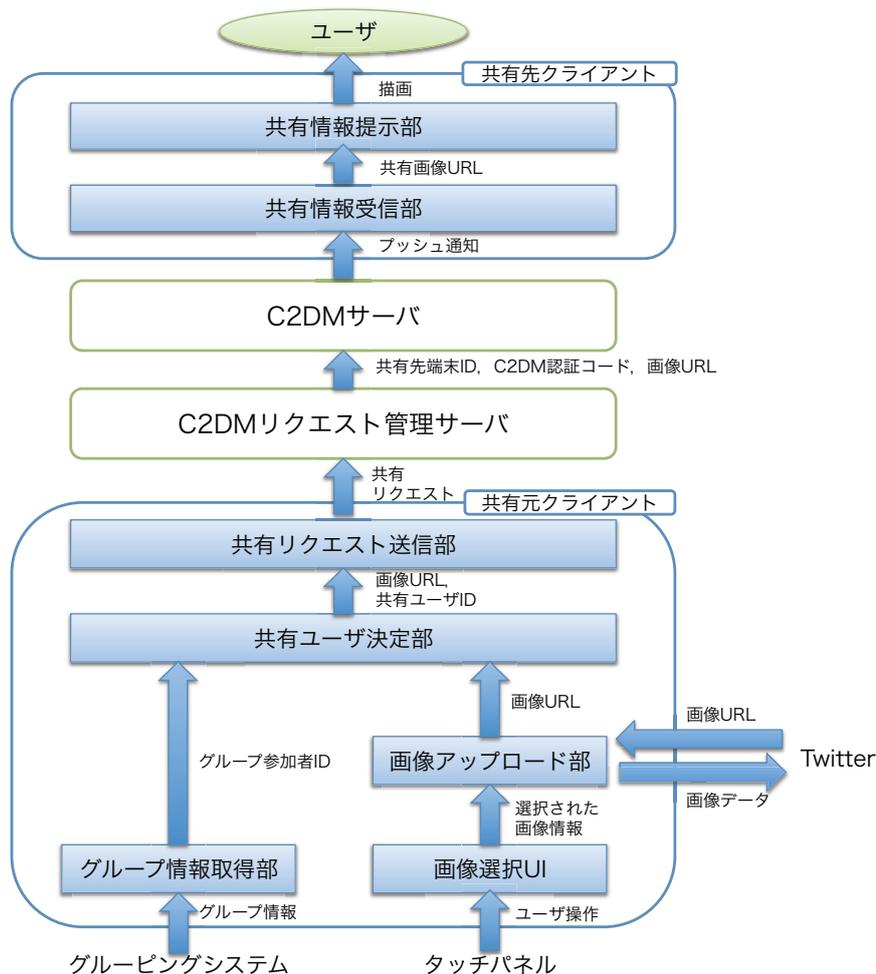
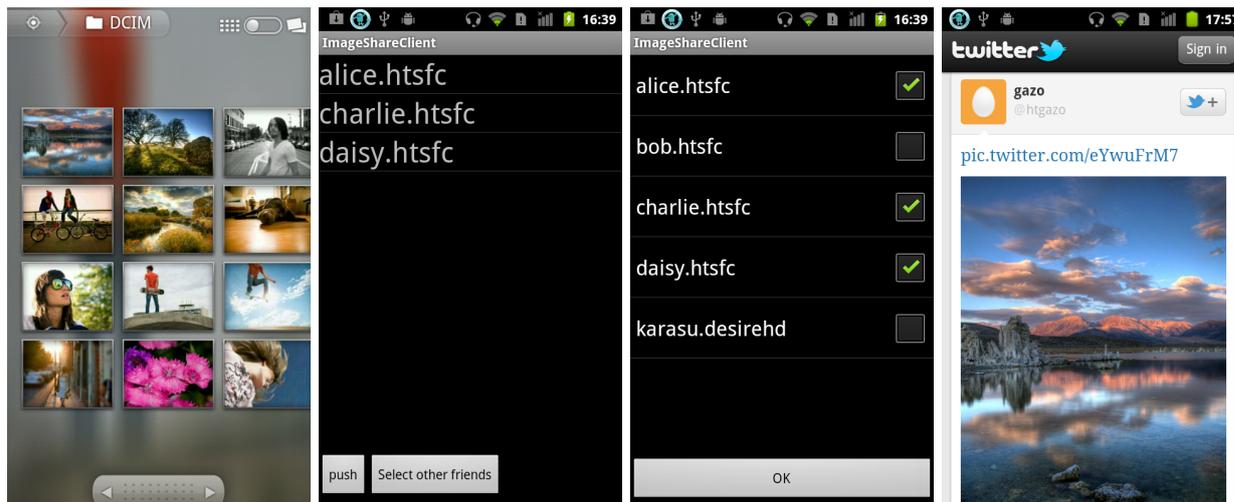


図 5.1 アプリケーション構成図

まず、試作アプリケーションは共有対象として選択された画像を一度 Twitter の画像アップロード機能を使用して Web 上に公開する。共有した画像の URL は Twitter のソフトウェア API によって試作アプリケーションに通知される。Twitter への画像アップロードは Android アプリケーションの実装言語である Java 言語で利用可能な Twitter API である Twitter4j を利用した。この URL をユーザと同じ会話グループに所属する人々の端末に Push 配信する。

URL の Push 配信は Google Cloud to Device Messaging (C2DM) フレームワークを使用して行う。C2DM フレームワークは Android 端末が Google アカウントと 1 対 1 で関連付けられている点を利用し、Google アカウントを指定することによって、そのアカウントに関連付けられる端末へとメッセージを Push 配信するフレームワークである。試作アプリケーションは、画像情報が Web に公開されて共有の準備が整うと、グルーピングシステムからグループに参加するユーザの ID すなわち Google アカウントを取得する。その Google アカウントに対して順に画像の URL を Push 配信するように専用の画像共有サーバへとリクエストを送信する。そして画像共有サーバから C2DM サーバへと Push 配信リクエストが送信されることにより、画像 URL は他の参加者が所持する端末へと順次配信される。



(a) 画像選択インターフェイス (b) グループ参加者表示画面 (c) 共有対象修正画面 (d) 共有結果画面

図 5.2 試作アプリケーションスクリーンショット

C2DM サーバから配信された画像 URL は各参加者の端末に配信された後、Android OS によって端末上で動作中のアプリケーションへブロードキャストされる。このブロードキャストは試作アプリケーションの画像表示機構によって受信され、同機構によってサーバから配信された情報から画像 URL が取り出される。最終的に端末のグローバル設定にて登録されているブラウザアプリケーションを試作アプリケーションが自動起動し、画像 URL にアクセスする。共有されたユーザは図 5.2(d) に示すブラウザ画面を通じて画像を閲覧することができる。以上のような処理によって画像情報の会話グループ内での共有が行われる。

端末アプリケーション実装言語	J2SDK 1.6.0
端末アプリケーション実行環境 VM	DalvikVM
UI 構築 API	Android SDK 2.3.4
Push メッセージ配信フレームワーク	Google Cloud to Device Messaging(C2DM)
画像アップロードサービス	Twitter.com
画像アップロード API	A Java library for the Twitter API(Twitter4j)

表 5.1 試作アプリケーション：クライアント実装環境

実装言語	Python version 2.5.5
実行環境	Google App Engine for Python
SDK	Google App Engine SDK 1.6.2

表 5.2 試作アプリケーション：サーバ実装環境

さらに試作アプリケーションにはグルーピングシステムが誤動作起こし、万一正しいグループ情報が得られなかった際のために、ユーザが共有相手の確認及び修正を共有事前に行うことのできる機能が実装されてい

る。試作アプリケーションは図 5.2(b) に示すように、ユーザが共有する画像を選択すると、グルーピングシステムから得られた会話グループの参加者の ID を表示する。そして、共有したくない相手がいる場合やグルーピング処理のエラーによって共有したい相手が欠落している場合に、ユーザは位置情報に基づいた近傍ユーザの ID の中から共有先を選択することもできる。図 5.2(c) に選択画面を示す。これにより、万一グルーピングシステムが誤作動を起こして、正常なグルーピングが行えない場合であっても、ユーザの負担を軽減しながら画像共有を完結することができる。

本研究では上述した実装構成で、情報に基づくコンテキストウェアサービスの一例として画像情報共有アプリケーションを試作した。

5.3 本章のまとめ

本章では、本研究が提案するグルーピングシステムがもたらす会話コンテキストの応用分野とその可能性についてまとめた。会話コンテキストはコンテキストウェアシステムにおけるサービス提供対象の指定、検索システムでの利用を想定した仮想情報に付与する注釈、そして実世界コミュニケーションの電子化のための基礎データとして応用が期待される。この議論を踏まえた上で、さらに会話コンテキストに基づくコンテキストウェアシステムの一例となる試作アプリケーションの実装について説明した。試作アプリケーションは会話コンテキストに基づいて、情報の送信対象の指定を行う情報共有アプリケーションである。

次章では 4 章で説明したグルーピングシステム、及び本章で示した試作アプリケーションを用いて行った評価実験に関してまとめる。

第 6 章

評価

次章までに説明した会話グルーピングシステムと会話コンテキスト情報に基づくアプリケーションを使用して、本研究の提案アルゴリズムの妥当性に関する評価実験を行った。本研究で行った評価実験と結果、そして実験結果に基づいた考察について説明する。

6.1 Ground Truth に基づくグルーピング精度評価

本研究では、Milvus のグルーピング精度について評価実験を行った。評価実験は音声解析部のエラーによるグルーピング精度の低下を評価から排除するため、被験者の会話音声に基づいて手作業で作成した発話区間情報に基づいてグルーピング処理を行い、その結果を Ground Truth（正解情報）と比較するという実験手法を取った。

6.1.1 手順

実験手順を説明する。本実験では被験者をセミパブリックな空間に集め、実験時間中その空間で自由に過ごすように指示した。その際、被験者には 5 章で示したスマートフォンに付属の通話用集音マイクを取り付け、内部システムの時刻を端末間で合わせた上で、端末を身につけてもらった。集音マイクは被験者の胸元に取り付け、被験者毎に会話音声を取得できる状態にした。さらにスマートフォンで集音マイクから取得した音声を録音するアプリケーションを起動し、実験中被験者の会話を録音する状態で実験を開始した。

実験中、被験者は自由に行動した。被験者は自分の PC に向かって作業をする、飲食をする、読書をするといった行動を行った。その中で被験者間で懇談をしたり、突発的なディスカッションを行ったりといったアドホック型会話を行った。被験者は自由に会話を行ったため、しばしば同時並行的に会話グループの発生が見られた。

実験終了後、録音した音声から被験者毎に、その被験者が発話を開始及び終了した時刻を発話を行うたびに記録し、実験時間内における被験者の発話区間情報を作成した。記録作業は実験に参加した被験者とは別の学生数人で行った。最終的に 1 回の実験で得られる発話区間情報は被験者の数だけ得られた。

実験後、発話区間情報に基づき、4 章で示した会話グループ取得機構によってグループ解析を行った。解析時の単位時間は 30 秒に設定した。これは予備実験の過程において、グルーピング精度とグループ情報の更新レートのバランスを考慮した結果設定した値である。解析後、単位時間毎に解析によるグルーピング結果と、会話グループの正解情報と比較し、その整合率を評価した。

6.1.2 実験環境と装置

実験は慶應義塾大学徳田研究室 Delta S103 号室にて行った。実験環境には実験中の光景を撮影するカメラを2台設置し、撮影した映像は実験後の正解情報作成時に利用した。1回の実験は約20分間行われ、参加した被験者は同研究室に所属する男女学部生6人であった。実験は全部で7回行われた。実験時に使用した端末、及び会話グループ取得機構を動作させた機器は4章において実装環境として示した機器と同型のものを使用し、ソフトウェアの実行環境も同環境で行った。図6.1に実験中の光景を示す。



図 6.1 S103 号室での実験の様子

6.1.3 結果

本実験の結果を図6.2に示す。E1からE7は各実験実施時の整合率を示し、右端の値は全実験の平均整合率を表している。グラフが示す整合率は次式によって算出された各グループの整合率の平均値によって表される。

$$P(g) = 100.0(|\text{Agree}_g| / (|R_{\text{result}_g} + C_{\text{correct}_g}|)) \quad (6.1)$$

数式中、 C_{correct_g} は正解情報として定義されている、グループ g に実際に参加していた被験者を表し、 R_{result_g} はグループ g の解析結果で参加していたと解釈された被験者を表す。つまり $|R_{\text{result}_g} + C_{\text{correct}_g}|$ は正解情報と解析結果のそれぞれの参加者を論理和した時の人数である。さらに、 $|\text{Agree}_g|$ は解析結果と正解情報を比較して正しくグルーピングされていた被験者の人数を表している。すなわち、数式6.1は本来のグループ参加者のグルーピング漏れ及び非参加者の誤検知を非整合とした場合のグループ毎の整合率を算出する。グラフでは実験時間中に発生した全グループの解析整合率の平均値を示している。なお、正解情報と解析結果のグループの関連付けは正解情報に定義されたセンタユーザを基準にしている。

各実験時間において発生したグループは、グループ参加人数が1人のみであるグループを含めて、平均で125グループであった。これは平均すると各単位時間においては約3グループ発生したということである。

グラフから分かる通り、提案手法によるグルーピングの整合率は最高で87.3%、最低で67.6%、平均して77.6%の整合率でグルーピングすることができた。さらに、会話グループに内包すべき被験者が適切にグルーピング出来なかった事例は平均して約15%、会話グループに無関係な被験者がグルーピングされてしまった

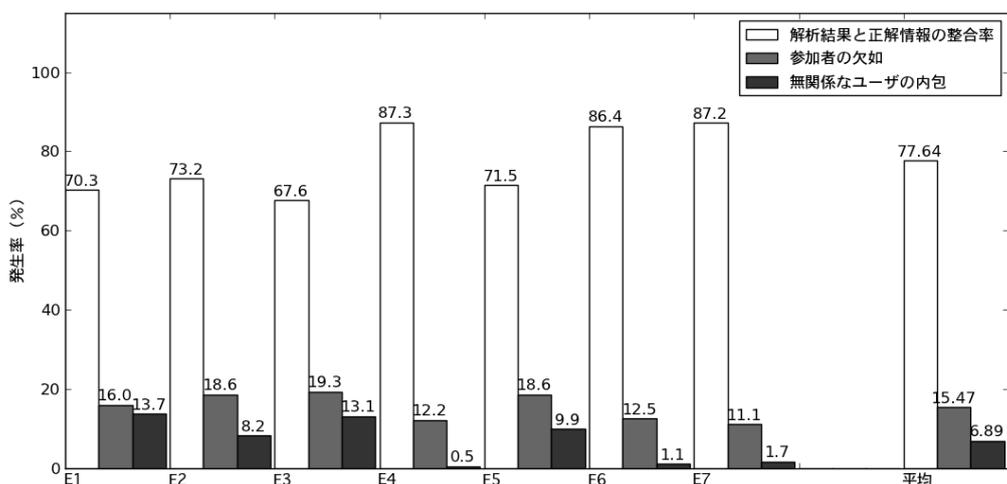


図 6.2 Ground Truth に基づくグルーピング精度実験結果

事例は約 6% であった。この精度でのグルーピングエラーはアプリケーションの実装で補完することが十分に可能な範囲であるため、提案手法によるユーザグルーピングは本研究の目的に十分であると言える。したがって、提案手法の有効性を示すことができた。

6.1.4 考察

提案手法によってグルーピングエラーが発生した場合について考える。図 6.3(a) 及び 6.3(b) は実験 E1 回において、実験開始から 10 分間で観測された実際の会話グループと、その際にシステムが算出したグルーピング結果を示した図である。図中 A から F が示す記号は抽象化された被験者の名称を表す。また矩形の連結によって各被験者のグループへの所属状況を表す。例えば実験時間 0:00 から 0:30 間の Ground Truth 情報では被験者 A, B のグループ、被験者 C, D, E のグループ、被験者 F のみのグループが形成されていたことを示す。

この図にも示されている通り、全体の実験結果から観測できたグルーピングエラーは次の 2 種類に区分される。

- (a) 解析結果に示されるグループからの参加者欠如
- (b) 無関係な非参加者の内包

エラー (a) が発生し、グループに内包されなかった被験者はグルーピング結果において青色で彩色されている。またエラー (b) が発生し、無関係なグループに含まれてしまった被験者は赤色で彩色される。これらのエラーに対し、その現象の詳細と原因について説明する。

参加者の欠如

(a) 解析結果に示されるグループからの参加者欠如は、本来グルーピングされていなければならない会話参加者が解析されたグループに含まれていないという、参加者のグルーピング漏れである。このエラーの原因として考えられる現象として、グループ内での無発話参加者の存在、センタユーザ選別ミス、同グループ参加者間での同時発話者誤検知の3つが挙げられる。

- 無発話参加者の存在

会話参加者が無発話であったケースは、まず、会話への参加者の中で「聞き役」に徹した参加者が存在するという点があげられる。単位時間内において全く発話を行わなかった、もしくは発話が他の参加者よりも極端に少なかった場合、そのユーザとセンタユーザ間の会話応答が観測できず、グループから除外されてしまう。会話の中に時折相槌を返す程度しか発話しない聞き役の参加者がいるような場合、周囲の会話参加者よりもセンタユーザの応答者として観測される機会が減少し、結果的にグループから除外されてしまうという事例が発生する。

図 6.3(b) で青色の枠線で囲われた結果において、本ケースが原因となる参加者欠如が発生したことを示される。本ケースによるグルーピングエラーは単純に無発話参加者のみがグループから欠落する。したがって、図中青枠線の結果に青色矩形で示される被験者は、該当単位時間中において全く発話を行わなかったと言える。

提案手法は基礎データとしてユーザの発話情報を使用しているため、ユーザの発話量が十分でないことが原因となるグルーピングエラーは対処が困難である。そのため、音声情報以外の基礎データから発話していないが会話に参加している人物を抽出するような機構が必要となる。本研究ではセンシング端末としてスマートフォンを想定しているため、通常スマートフォンに搭載されている方位センサや位置情報取得機器、加速度センサから得られるセンサ情報を音声情報と合わせて収集可能である。これらのセンサ情報に基づいて提案手法によるグルーピングを補完することで対応できると考えられる。これに関しては、アドホック型会話に参加中のユーザ間でセンサ情報にどのような特徴が観測でき、グルーピングの基礎データとして利用可能であるか、十分に検証する必要がある。

- センタユーザの選別ミス

次にセンタユーザの選別ミスによる参加者欠如について考える。参加者欠如のグルーピングエラーの原因となるセンタユーザの選別ミスは同じ会話グループに参加するユーザの中から複数のセンタユーザを選別した場合である。センタユーザの選別ミスは、同じグループ内に高頻度発話者が複数存在した場合において発生する。本研究における高頻度発話者の選出は、単位時間毎に発話区間の比較を行うユーザの間で相対的に閾値を設定し、閾値に基づいて高頻度発話者とそうでないユーザ群に二分する。グループ間で参加者の発話量に大きく差がある場合や発話量の多いユーザ間で単位時間内の発話量が均衡した場合に、この閾値による二分法が適切に機能しない。これにより、同じグループに参加するユーザから複数のセンタユーザを選別してしまう。本提案手法ではセンタユーザは各会話グループに1人のみ存在すると解釈するため、結果的に本来の会話グループを分割した状態でグルーピングが行われてしまう。図 6.3(b) においては、本事例によるグルーピングエラーが起こった結果を緑色の枠線で囲って示す。本事例によるグルーピングエラーが起こると、しばしば本来のグループが誤選別されたセンタユーザに沿って分割された状態でグルーピングされる。図中 2:30 から 3:00 のグルーピング結果では、被験者

A, B, D, F が形成したグループが A, B と D, F に分割されて 2 つのグループとして解釈された。これは当該グループの中からセンタユーザが複数選別されたため、グループの分割が起こったと言える。この他、センタユーザの選別ミスによって会話応答頻度算出時に応答者の食い違いを併発し、上述のエラー (b) も同時に起こった事例が図中 5:00 から 5:30 のグルーピングにて発生している。被験者 A, B, F で構成されるグループからセンタユーザが複数選別され、その結果グループが分断し、さらに無関係な被験者 E がグルーピングされている。このように本事例によるグルーピングエラーが観測された。この事例に対して、高頻度発話者からのセンタユーザの選定の条件として過去の単位時間におけるグルーピング結果を組み込むことが対処機構として考えられる。過去の単位時間において同じグループの参加者であると複数回解析されているユーザ同士が両方センタユーザとして選別されうる場合、その単位時間においては片方のユーザのみをセンタユーザとするという補正方法が考えられる。またセンタユーザ候補とされるユーザの内、他の候補がそれまでの解析結果で同じグループに属していなかったユーザと同グループに所属していた候補者を、他の候補者と異なるグループのセンタユーザとするといった対策も同様に考えられる。これらのような過去のグルーピング結果からパラメータを補正するアプローチは、連続してグルーピング処理を繰り返すとグルーピングエラーが蓄積していき、正常なグルーピングを阻害する恐れがある。そのため本論文において、過去のグルーピング結果に基づいてグルーピング処理を補完する機構は採用しなかった。今後センタユーザの選別ミスによるグルーピングエラーを補完するために、グルーピングエラー蓄積時の修正方法も合わせて組み込みながら、機構の追加を検討する必要がある。

- グループ参加者同士の同時発話者関係の成立

最後に同グループ参加者同士が同時発話者関係にあるとシステムが誤解釈したことによるグルーピングエラーは、同じ会話に参加するユーザの発話が他のユーザと比較してオーバーラップし続けた結果、会話中の相手を同時発話者と誤認識したことによって互いに異なるグループに参加していると解釈されてしまったケースである。例えば、会話中の相手の発話に被せるように発話を開始することが多いユーザが存在する、会話相手の発話の途中で頻繁に相槌を行ったユーザが存在する、会話グループ内の参加者が一斉に笑い出す、といった場合において同じグループの参加者間で同時発話者関係が成立してしまう。

図中では赤色の枠線で囲われた結果において、本事例によるグルーピングエラーが発生した。5:30 から 6:00 間グループ解析では被験者 A, B の発話が互いにオーバーラップし合い、同時発話者定義の生成過程において A と B が同時発話者として定義付けられた。これにより被験者 A, B は互いに別グループに所属すると誤検知された。さらに 8:00 から 8:30 のグルーピング処理では被験者 C, D, E のグループにて盛んな談笑があり、それによって 3 人の発話が他の被験者と比較して大きくオーバーラップしたため、被験者 C, D, E がそれぞれ同時発話者として定義された。これによって、当該グループは各被験者の単一参加者のグループとして解釈されてしまった。

これらの事例に関して、センタユーザ選別ミスによるグルーピングエラーと同様、過去の解析結果に基づいて同時発話者選定処理の補正を行うという対処方法が考えられる。具体的には同時発話者選定時、過去の単位時間において頻繁に同じグループとしてグルーピングされていたユーザ同士は選定に使用するオーバーラップ量に補正係数をかけるといった対処法により補正が可能である。

以上の議論から、参加者がグルーピング結果から欠如するエラーは音声情報以外のセンサ情報と過去のグルーピング結果に基づいて、高頻度発話者や同時発話者の選定処理に補正処理を加えることによる対処の効果が期待できる。ただし、これらのエラーは多くの場合、部分的にはグルーピングに成功しているため、欠如したユーザの補完インターフェイスをアプリケーション側で実装することでも、実用レベルの対処が可能である。

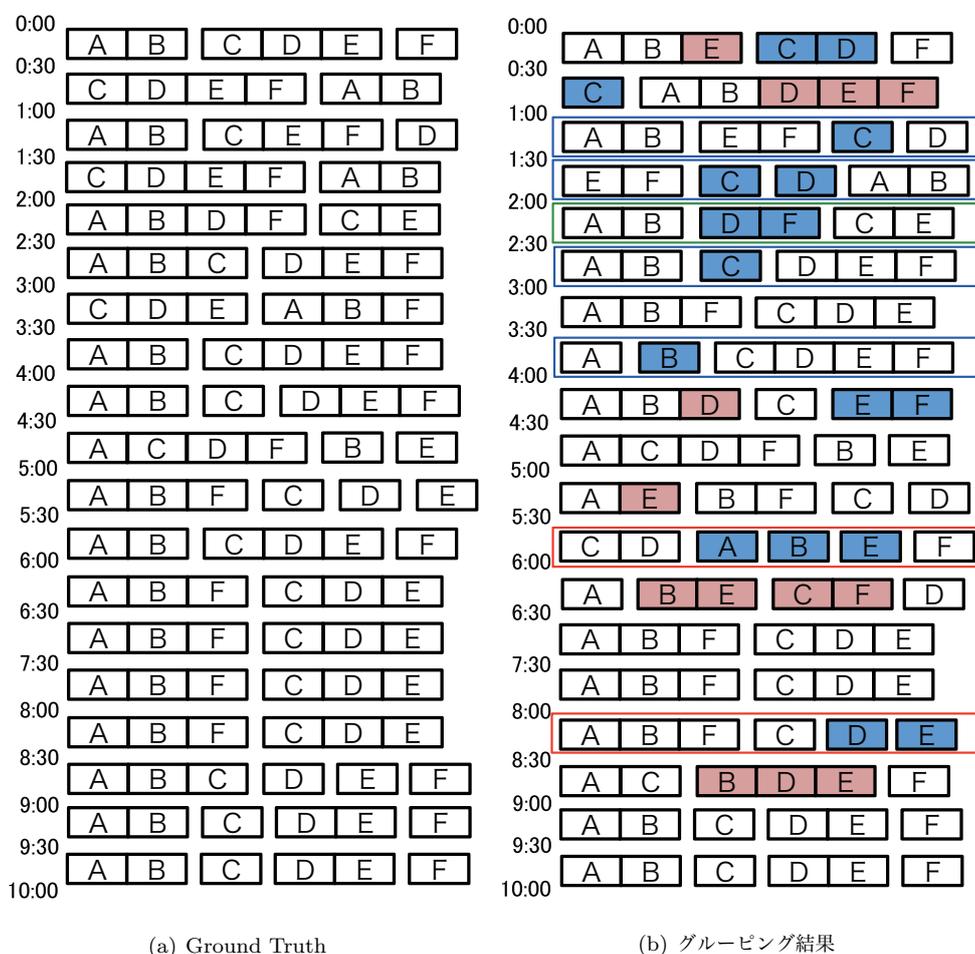


図 6.3 実験 E1 実験結果内訳抜粋：実験時間 0:00 から 10 分間の Ground Truth 情報とグルーピング結果の比較

無関係なユーザの内包

(b) 会話グループに会話に参加していない無関係なユーザを誤って内包してしまうエラーは、会話グループが同時に複数発生し、尚且つ各会話グループ内で発話者が頻繁に入れ替わるような熱心な会話が行われた場合に発生した。このエラーの原因は会話応答頻度算出時の応答者の食い違いである。あるセンタユーザに対する会話応答頻度算出時に本来の応答者ではなく、別グループの発話者を応答者であると誤検知する事例が多く発生した場合において、グループ参加者の入れ替わりが発生した。

この応答者の食い違いは提案手法におけるセンタユーザの選定と周辺ユーザの限定処理によって、会話応答頻度を総当たりで算出するよりも軽減されていると考えられる。エラーの原因となる応答者の食い違いは、同

一空間内で会話を盛んに行うグループが複数発生した場合に発生したためである。このような事例において応答者の食い違いの発生を抑制するため、センタユーザ選定と周辺ユーザの限定アルゴリズムや会話応答頻度の算出アルゴリズムの改良による精度向上が不可欠である。具体的には会話応答者の選定において、現アルゴリズムで採用している会話開始時刻の最低差分者を応答者として解釈するのではなく、応答者と非応答者の発話開始タイミングに関する統計的な差分を明らかにし、統計的な知見に基づいた応答者の判別を行うようにアルゴリズムを改良する方法が考えられる。

ところでこのエラーは無関係な人物が会話グループに内包されてしまうという、本研究の機能要件の1つとして定義された「近傍非参加者のグループからの排除」で達成すべき課題が満たされない事例である。本エラーの発生頻度は平均して約6%であった。これに関して、グルーピング結果を利用する際にアプリケーション側においてグループ情報修正機構を準備することで柔軟に対応可能だと考えられる。本研究で試作したアプリケーションにおいても、位置情報から粗粒度グルーピングした結果に基づいてグループ情報を修正する機能が備えられている。実際に本エラーが起こった場合、アプリケーション利用者が必要にあわせてグループ情報を修正することによって、本研究が想定する情報共有システムや情報検索システムようなアプリケーションにおいては対応可能であると考えられる。したがって、本研究が想定するグループ情報の応用範囲には、提案手法は有効であったと結論づけられる。

提案手法の有効範囲

以上のグルーピングエラーとその原因に関する考察を受けて、提案手法が有効的に動作する手法の適応可能環境について考察する。

本実験の結果から考察できることは、単位時間内に発生したグループの数量が増加すると正常なグルーピングが妨げられるということである。したがって、ユーザが多く集まる広域パブリック空間のような場所では正常なグルーピングが期待できない。例えば、多数の通行人が一度に集まる道路交差点上で自然発生した会話のグループをグルーピングするような場合は正常なグルーピングは困難である。したがって、大量のユーザが一箇所に集まるような場合では、位置情報に基づく粗粒度グルーピングをより細粒度で行うことや、音声情報だけでなく他のセンサ情報やソーシャルネットワークサービス(SNS)上で公開される友人関係情報等の基礎データを利用してグルーピングするといったグルーピング手法が好ましい。

合わせて、飲食店内のように非常に雑音が多い環境下においては、ユーザの発話区間情報抽出を行う音声解析処理部に雑音環境にロバストな解析手法を組み込む必要がある。4章で説明した音声解析部は雑音環境に弱い。そのため、雑音環境におけるグルーピングは音声解析部の発話取得誤差に依拠して、精度が低下する。しかしながら、提案手法は音声処理結果を基板としているが、音声解析処理そのものに依存したアルゴリズムではない。そのため、センシング端末のクライアントシステム側で周囲の環境によって音声解析アルゴリズムを切り替えるといった手法によって対処が可能である。既存研究では、Emiliano 氏らの DarwinPhone[2] の様にスマートフォンをセンシング端末とした発話者認識において、雑音環境下においてもロバストな認識を実現するアルゴリズムが提案されているため、音声解析部をよりロバストな手法に換装することによって提案手法でも十分に対処可能であると言える。

提案手法が最も有効に働く環境は、これらのような環境ではなく、中規模のユーザがグルーピングの母集団となるセミパブリックな環境である。例えば、企業のオフィス内や大学構内の研究室等が挙げられる。これらの環境は本実験の実験環境と酷似している。本実験結果においてこのような環境下におけるグルーピングは有効であるということが出来、さらに本研究で想定するアプリケーションシナリオのようなニーズがよく発生する環境でもある。したがって、提案手法は実験環境のように中規模のセミパブリックな屋内環境において、最

も効力を発揮する。

提案手法の計算量に関する考察

本提案手法においてユーザ数が増加していった場合のグループ情報計算コストについて考える。グルーピングシステムをデプロイする場合、会話グルーピングの対象となるユーザが増加した時に Milvus のスケールについて考慮する必要がある。本実験はごく少人数の被験者を起用して実施したため、本実験では計算コストを測定する上で正確な知見は得られない。しかし、提案手法の位置情報による粗粒度グルーピングによってグルーピング処理以前にフィルタリングすることで、演算対象を小集団に区分できるため、計算爆発は防止できると言える。位置情報によるフィルタリングはリレーショナルデータベースの機能を用いることによって、よりフィルタリング処理コストを低減することが期待でき、処理全体の計算コスト低減に有効であると考えられる。

ただし、実際にグルーピングシステムをサービスとして稼働するためには、ユーザ数が増加しても高速処理が可能なシステムの構築が必要である。本研究ではグルーピングシステムは遠隔サーバシステム上で稼働されることを想定している。そのため、ソフトウェアの実装による高速化の他に、位置情報によって区分したユーザ群を並列にグルーピング処理できるように大規模な並列処理環境を整えることや、ユーザの増加に合わせて演算環境をスケールアウトできるハードウェア構成を構築するといった対策が考えられる。このように、大規模にサービスとして展開する場合は高速なグルーピングを可能とするため、高速処理のためのチューニングが必要である。

6.2 試作アプリケーションによるユーザビリティ評価

次に、本研究で想定とするアプリケーションシナリオにおいて、提案手法によるユーザグルーピングの有効性を示すため、5章で説明したアプリケーションに基づいたユーザビリティ評価を実施した。ユーザビリティ評価実験は本研究で想定するアプリケーションシナリオに基づき、会話中の相手を指定して携帯端末を介して情報を送信するというシーンを再現し、情報送信相手の指定において会話コンテキスト情報の利便性を他インタラクションと比較評価する。具体的には、会話相手全員に携帯端末内の画像ファイルを送信する際において、電子メールアドレスによる送信対象指定、物理的なインタラクションによる指定、会話コンテキスト情報に基づく指定の3手法のユーザビリティを評価した。

6.2.1 実験計画

1 要因 3 水準被験者内実験計画である。第 1 要因は画像送信相手の指定方法（電子メールアドレスによる指定、物理インタラクションによる指定、会話コンテキスト情報に基づく指定）である。従属変数は画像送信が完了するまでに要した操作時間と実験後に被験者が回答するアンケートの各回答項目である。電子メールアドレスによる指定時の画像送信は、Android 標準の Gmail クライアントアプリ上で画像添付メールを送受信することによって行う。また物理インタラクションによって端末をペアリングして情報共有するアプリとして Bump クライアントアプリ [3] を利用する。会話コンテキスト情報に基づく画像送信には 5 章で説明したアプリケーションを利用する。各水準間でバイアスがかからないよう、被験者毎にカウンターバランスをとって実験を実施する。

6.2.2 実験環境と装置

実験環境は慶應義塾大学湘南藤沢キャンパス 7 館 22 号室で行った。実験環境には被験者の他に、実験監督者が同席して被験者に実験進行に関する指示を行った。

被験者が身に付けるセンシング端末は前節の実験で使用した端末と同様のセンシング端末を利用した。端末内には 5 章で説明した音声解析部が動作するバックグラウンドアプリケーションが実行されており、マイクで取得した音声情報から被験者の発話区間を解析した。また、その発話区間情報は一定時間毎に端末位置情報とタイムスタンプと共にリアルタイムグルーピングサーバへと送信した。端末は実験前に時計合わせを行って使用した。

端末から送信された発話区間情報に基づいてユーザグルーピング処理を定期的に行い、同時にリアルタイムのグルーピング結果をアプリケーションに配信する解析サーバを設置した。サーバのハードウェア環境は 4 章で示したサーバ環境と同様であった。実験中グルーピング処理に対するノイズを発生させるため、被験者の発話区間情報と共に 2 人分のダミー発話区間情報を解析サーバに送信して、被験者の発話区間情報と合わせてグルーピング処理を行った。なお、ダミー発話区間情報は事前に 2 人の学生が実際に会話をを行い、その会話音声解析することで得られた発話区間情報である。またグルーピング処理の単位時間は 30 秒、クライアントソフトウェアでの音声入力は PCM 16bit エンコーディングのモノラルチャンネルで 4096Hz で行い、音声解析に用いるパラメータは振幅閾値 2000、零交差数閾値は 2、1 フレームの離散データ数は 512、すなわち発話区間は 0.125 秒刻みに算出するように設定した。

6.2.3 手順

実験は 3 人 1 組で行った。まず被験者は各自センシング端末を渡され、集音マイクを胸元に取り付けるよう指示された。そして、実験監督者が各アプリケーションにおける画像送信方法を口頭で被験者に説明した。

次に、被験者の内 1 人が他の被験者に対して端末内に保存されている画像を各アプリケーションを利用して送信した。その際、各アプリケーションの操作時間を実験監督者が記録した。操作時間はアプリケーションの操作を開始してから他の 2 人被験者に画像の送信が完了するまでの時間である。さらに同様の実験を他の被験者も繰り返して行った。

被験者全員がこの実験手順を終了した後、端末を回収し、被験者にはアンケートの回答項目に記入してもらった。

6.2.4 被験者

本実験に参加した被験者は 18 歳から 23 歳までの男性 11 名、女性 4 名、合計 15 名であった。全被験者は日常的に携帯端末を所持し、利用している人物であった。

6.2.5 結果

本実験の結果を以下に示す。

必要操作時間の比較

各指定方法の操作時間結果を表 6.1 に示す。比較対象の中で平均操作時間が最も短かった手法は本アプリケーションを用いた会話コンテキスト情報による対象指定であった。会話コンテキストによる手法は最大操作時間、最小操作時間ともに他手法と比較して短かった。この結果は既存のサービス対象指定方法に比べ、会話コンテキスト情報に基づいたコンテキストアウェアシステムが利便的であることを示している。さらにこの結果は、ほぼ全ての被験者が本アプリケーションを素早く使用することができたこと、及び提案手法によって解析されたグルーピング情報が実際にアプリケーション上で利用可能であるということを示している。

手法	最大操作時間	最小操作時間	平均操作時間
電子メール	6分56秒	1分23秒	3分9秒
物理インタラクション	3分46秒	30秒	1分8秒
会話コンテキスト情報	1分37秒	9秒	34秒

表 6.1 操作時間の比較結果

操作負荷の比較

実験時に各手法によるサービス対象指定について、最も負荷を感じなかった手法、及び最も負荷を感じた手法のそれぞれを1つずつ選択してもらった。その結果を表 6.2 に示す。多くの被験者が会話コンテキスト情報に基づく手法が最も低負荷であったと回答した。さらに同手法に対して最も負荷を感じた被験者は皆無であった。以上の結果から、会話コンテキスト情報はより利便性の高いサービスの構築を可能にしたと言える。また、本研究が想定するアプリケーションにおいて提案手法は有効であると示している。

設問	電子メール	物理インタラクション	会話コンテキスト情報
対象指定の負荷が最も低かった	0	5	10
対象指定の負荷が最も高かった	14	1	0

表 6.2 送信対象指定時負荷の比較結果

会話コンテキストの有効性

さらに Milvus が解析するような会話コンテキスト情報の利便性、妥当性について、被験者に表 6.3 に示す設問に対して4段階の点数で回答してもらった。各設問は得点が高くなるほど設問の文言に被験者が同意していることを示す。同表に回答結果を示す。「会話コンテキスト情報は携帯端末サービスにおいて有用である」、「会話コンテキスト情報を実際に利用するコンテキストアウェアシステムを利用したい」という2つの設問に対してほぼ全ての被験者が3以上の点数を回答した。このことから Milvus によって解析、配信される会話コンテキスト情報はコンテキストアウェアシステムに置いて有用であることを示すことができた。

設問	1	2	3	4	平均スコア
会話コンテキスト情報は携帯端末サービスにおいて有用である	0	0	10	5	3.33
会話コンテキスト情報を実際に利用する コンテキストウェアシステムを利用したい	0	1	10	4	3.2

表 6.3 会話コンテキスト情報の有効性

6.2.6 考察

本実験結果に基づき、本研究が想定するシナリオにおける課題を解決する上での Milvus の有効性について考察する。表 6.1, 6.2 に示す結果より、Milvus による会話グルーピング結果は既存のアプリケーションよりもユーザにとって低負荷に利用可能なコンテキストウェアシステムの構築に有効であると示すことができた。特に他アプリケーションよりも操作時間が圧倒的に短時間であるという点は、グループ内で進行中の会話のインタラプトを低減することができ、会話の最中であっても利便性の高いシステムであると言える。

本研究が想定したシステムの利用環境は、オフィスルームといった屋内のセミパブリック空間である。本実験環境はこの想定環境に則した環境であるため、本研究が想定するシナリオにおいて、Milvus は十分な有効性がある。しかし、広域パブリック空間のような環境におけるグルーピングは提案手法では対処困難であるため、会話コンテキスト情報抽出可能範囲の拡大が必要である。そのため、別途同環境におけるユーザグルーピングアルゴリズムについて検討し、ユーザの現在位置に応じてグルーピングアルゴリズムを切り替える機構の構築といったことが今後の展望として挙げられる。

表 6.3 で示した会話コンテキスト情報の妥当性、利便性に関する設問について、被験者の中で会話コンテキスト情報を利用するコンテキストウェアシステムを利用したいか、という設問に対して 2 点と回答した被験者が 1 人だけ存在した。この被験者に対し、別途回答理由についてヒアリングを行ったところ、当該被験者がアプリケーションを操作時にうまくグルーピングできていなかったことを理由として挙げた。これに関しては前節の評価考察で説明したとおり、アルゴリズムの精度向上が必要であると言える。ただし、ほぼ全ての被験者が会話コンテキスト情報は妥当性、利便性があると回答したことから、Milvus による会話コンテキスト情報はコンテキストウェアシステムにおいて十分な妥当性と利便性が得られたと言える。

以上の評価結果から、Milvus による会話グループのグルーピングは本研究の想定するシナリオにおいて、既存のシステムよりも利便的で低負荷なコンテキストウェアシステムの構築に役立ったと示すことができた。したがって、Milvus が解析を行った会話コンテキスト情報は本研究が想定するアプリケーションの実現に十分に有効性があると言える。

6.3 本章のまとめ

本章では本研究が提案する会話グルーピングシステムの有効性を示すために実施した評価実験とその結果、考察について説明した。その結果、提案手法によるグルーピングは本研究が想定するシナリオにおいて十分な有効性があることを示すことができた。

これらの結果、考察に基づき、次章で本研究をまとめる。

第7章

結論

本章ではこれまで述べた内容に基づき、本論文の結論を述べる。まず本研究における今後の展望について説明した上で、本論文をまとめる。

7.1 今後の展望

本論文が提案したアドホック型会話におけるユーザグルーピング手法に関して、今後改善すべき点、拡充すべき点について述べる。

7.1.1 多様なセンサ情報の利用による精度向上

Milvus の改良すべき点として、スマートフォンが一般的に搭載するセンサを用いたグルーピング精度向上が挙げられる。本論文が提案するグルーピング処理は、ユーザの音声情報という対ノイズ性能が極めて低いセンサ情報に基づいてグルーピングを行う。そのためシステムの使用環境の差分、周囲の雑音や集音マイクの有無によって、グルーピング精度が影響を受けやすいと考えられる。

本研究においてはセミパブリック空間での利用を想定した精度評価実験を実施し、ある程度の実用に耐える精度を得ることができた。しかし、今後多様なコンテキストウェアシステムにおいて会話コンテキスト情報を応用可能にする必要があることや、セミパブリックな空間だけでなく、広域パブリック空間においても同様にグルーピングを可能にする必要があることは事実である。以上の理由からさらなる精度向上は不可欠である。

これについて、センシング端末として想定するスマートフォンは多様なセンサを内蔵していることから、これらのセンサ情報に基づいてグルーピング処理を補完することによって精度向上を図ることができると考えられる。スマートフォンには方位センサや加速度センサを始めとする多様なセンサが内蔵されている。これらのセンサ情報に基づいてユーザの行動認識を行い、その情報に基づいてアドホック型会話へ参加中のユーザの絞り込みや Milvus のグルーピング処理におけるセンタユーザ及び周辺ユーザ選出処理の改良を行う。

この改良策の方針として、まずはアドホック型会話中の人々の行動に関する予備データを収集することが第一である。アドホック型会話のグルーピングにおいて利用価値の高い人々の行動傾向について精査する必要がある。例えば、「アドホック型会話を行なっている人々は互いに体が向かい合うことが多い」といったような、アドホック型会話参加者にまつわる行動傾向に関しての情報を収集する。このデータの中からデータマイニングを行い、グルーピング処理に利用価値の高い行動認識情報を見つけ出すことを第一に行わなければならない。

したがって、本展望についてはまず、多様な環境において人々のアドホック型会話に関するセンサ情報の収集を行い、同時に人々の行動傾向のデータ化を実施することが先決である。

7.1.2 定性的情報に基づくグルーピング処理の補完

スマートフォンに搭載されるセンサから得られる定量的な情報だけでなく、ユーザの交友関係情報や趣味趣向に関する情報などの定性的な情報によるグルーピングの補完を上述した改善点と合わせて実施することが求められる。

Milvus によるグルーピング処理が適応可能な環境を拡大していくにつれ、会話コンテキスト情報の取得粒度について考える必要がある。すなわち、どこまで無関係な人々とのグルーピングを許容するか、各ユーザが決定可能である必要がある。例えば、日常生活内では、普段から面識のある人々との会話から商店での売買時に行われる店員との会話のように全く面識のない人々の会話も頻繁に発生する。中には自分の友人の友人を交えた会話のように、間接的に面識のある人々との会話等中間的な位置づけの会話もありうる。これらの会話に対して、ユーザプライバシーやユーザの趣向の観点から、どこまで会話コンテキスト情報として抽出を行うかについてユーザ側で設定が可能である必要がある。

これに関して、ソーシャルネットワークサービス (SNS) をはじめとする Web サービスから得られる情報を利用することで実現可能であると考えられる。SNS にはユーザの交友関係や趣味嗜好といった個人情報が登録されている。特にユーザの交友関係に関する情報は会話コンテキスト情報の粒度を区分する上で有効であると期待できる。例えば、面識のない人々とのグループ形成を無条件に許容する他、SNS 上で交友関係がある人に限定してグルーピングを行い、面識のない人々とのグルーピングは除外するような方法が可能となる。さらに、このグルーピング許容範囲の限定は交友関係のホップ数に基づいてユーザやアプリケーションが決定するという方法も考える。交友関係情報を利用することで、“友人のみ”でグルーピングする場合や“友人の友人”までの範囲をグルーピング対象として含めるといった設定をユーザがアプリケーション毎に行うことが可能である。このように交友関係に関する情報はグルーピングの対象範囲を区分することに利用可能であると考えられる。

以上のように、SNS から取得できる情報に基づいて提案手法のグルーピング処理の柔軟性を向上させることで、より利用価値の高い会話コンテキストの取得が期待できる。この定性的な情報に基づくグルーピングの補完は今後の課題と言える。

7.1.3 会話グループ情報取得インタフェースの充実

最後に、グルーピングシステムをより実用的に拡充するため、コンテキストアウェアシステムがグルーピングシステムから会話コンテキスト情報を取得するためのインタフェースをより充実する必要がある。

本論文で実装したグルーピングシステムには、算出した会話コンテキスト情報を取得するための最低限のインタフェースが実装されている。このインタフェースはユーザ ID をキーとして問い合わせると、そのユーザが所属する会話グループ情報を返すという動作を行う。本グルーピングシステムによる会話コンテキスト情報が様々なコンテキストアウェアシステムで応用できるようにするため、インタフェースの充実させる必要がある。具体的には、ユーザ ID によるグループ情報の取得だけでなく、位置情報や時刻情報をキーとするグループ情報取得を提供するインタフェース等の実装が挙げられる。

位置情報をキーとするグループ情報取得は、最も単純なもので絶対位置情報によって地理的な範囲指定を行

い、指定した範囲で発生したグループ情報を取得するというインタフェースである。情報取得クエリとして緯度経度と半径を指定し、指定された緯度と経度を原点とする円領域と各グループ情報の位置情報比較した結果、円領域内で発生したグループ情報を取得するというインタフェースが考えられる。これにより、ショッピングモールや大学構内といったある特定の施設内のグループ情報を一括で取得する事が可能となり、特定施設内で発生したグループを対象とするレコメンドサービスのようなシステムが容易に構築できる。また時刻情報をキーとした取得インタフェースは、単純にある時刻区間に発生したグループ情報を取得するというものである。これは既存のデータベースクエリでも良く見られるような、グループ情報のタイムスタンプを基準としてグループ情報を一括して取得するというインタフェースである。したがって、このインタフェースでは比較基準となるタイムスタンプを指定するクエリを送信するという事が考えられる。これらのインタフェースはそれぞれ併用することによって、アプリケーション内の用途に則して柔軟な情報取得が期待できる。

以上のような、システム開発者に利用価値の高い取得インタフェースを提供するために、インタフェースプロトコルの設計は開発者のソースコード記述量の削減や開発工数の低減等開発労力を減少させることを意図して行わなくてはならない。また、スマートフォンのような移動端末において度々問題とされるネットワークの一時的な遮断を考慮し、ネットワーク通信に依存しすぎない構成での実装を行う必要がある。

以上のニーズを考慮した上で、グルーピングシステムの情報取得インタフェースの拡充を行う必要がある。

7.2 まとめ

センサデバイスやそれを内蔵する端末の進歩によって、これらのデバイスを介して取得できるセンサ情報に基づいたコンテキスト情報の取得技術の研究開発が盛んに行われている。このコンテキスト情報解析において、“いつ、どこで、誰と会話を行ったか”という、実世界で行われる会話に関するコンテキスト情報は様々なコンテキストアウトウェアシステムに応用が期待される。

会話コンテキスト情報の抽出にはコンテキスト解析システムが会話の発生を検知し、その会話の参加者を特定する、すなわち参加者のグルーピングを行う必要がある。本論文では実世界で人々がFace-to-Faceで行う会話の中でも、日常生活内において自然発生する「アドホック型会話」を対象として会話コンテキスト情報を抽出する。アドホック型会話は会話への参加者が少人数で、会話の発生場所、時刻が事前に予測困難な会話である。さらにアドホック型会話は友人との懇談や社内での同僚間の相談といった、日常生活内における会話の大部分を占める。

発生場所と時刻が予測できないアドホック型会話において、コンテキスト情報解析に用いる基礎データ収集は環境内に敷設されたセンサネットワークによる情報収集は効果が期待できない。したがって、ユーザ周囲の情報の常時取得によって行う必要がある。さらに、アドホック型会話は会話が行われるすぐ近傍に会話とは無関係な人々が存在する場合がある。そのため、会話コンテキスト情報解析システムは会話参加者と近傍の非参加者を区分できる必要がある。

本論文ではアドホック型会話の会話コンテキスト情報の抽出のため、人々の会話応答頻度に基づく会話参加者のグルーピング機構であるMilvusを提案した。会話応答頻度とは、“単位時間内において誰が誰の発言に何度応答したか”を表す。Milvusでは人々の会話応答頻度に基づき、頻繁に会話応答を行う人々を互いに同じ会話へ参加していると判断してグルーピングを行う。会話応答頻度算出時、Milvusは高頻度発話者や同時発話者といった人々の発話の特徴を掴み、その情報を基により信頼性の高い会話応答頻度算出を行う。これによってアドホック型会話における会話グルーピングを実現する。

Milvusを組み込んだ会話参加者グルーピングシステムを実装し、グルーピング精度についての評価実験を

行った。その結果、本論文が想定する利用環境において、平均して70%を超える精度でグルーピングが可能であることを示した。さらにグルーピングシステムによる会話コンテキスト情報に基づくコンテキストアウェアシステムを実装し、そのユーザビリティを評価した。その結果、Milvusによる会話コンテキスト情報はユーザビリティの高いコンテキストアウェアシステムの構築に有効であることを示すことができた。

本研究の今後の展望として、音声情報の他に多様なセンサ情報、ユーザの友人関係情報といった定性的な情報からグルーピング処理を補完することや、アプリケーションが利用する会話コンテキスト情報取得インタフェースを拡充することが挙げられる。以上の結果と考察に基づき、本論文の提案手法はアドホック型会話の会話コンテキスト情報の抽出の実現について貢献したとして、本論文のまとめとする。

謝辞

本研究の機会を与えてくださり、絶えず丁寧なご指導を賜りました、慶應義塾大学環境情報学部教授徳田英幸博士に深く感謝致します。また、貴重なご助言を頂きました慶應義塾大学環境情報学部准教授高汐一紀博士、慶應義塾大学環境情報学部教授清木康博士、慶應義塾大学環境情報学部専任講師中澤仁博士には本研究を進めるにあたって多くの励ましとご指導を頂きました。ここに深い感謝の意を表します。慶應義塾大学徳田研究室の諸先輩方には折に触れ貴重なご助言を頂き、また多くの議論の時間を割いて頂きました。特に政策・メディア研究科特任助教米澤拓郎氏、政策・メディア研究科研究員伊藤友隆氏、政策・メディア研究科博士課程生天目直哉氏、政策・メディア研究科修士小川正幹氏、今枝卓也氏には、本論文の執筆にあたってご指導を頂きました。ここに深く感謝いたします。また、大学院進学についての理解をいただき、研究生生活を支えてくれた家族に深い感謝の意を表します。

最後に、同じ研究室の学生として共に過ごした、慶應義塾大学徳田研究室天野雅哉氏、井村和博氏、星北斗氏、堀川哲郎氏、蛭田慎也氏、瀧本拓哉氏、研究グループ ACE 及び CPSF の学生、論文執筆にあたり多大な協力をしてくださった西條晃平氏、西山勇毅氏、勝治宏基氏、寺山淳基氏、伊藤瑛氏、小鷲麻奈美氏、坂村美奈氏、加藤昇子氏、そして、研究室修士課程同期として共に励まし、悩み、奮闘し、兄弟同然に研究の日々を過ごした政策・メディア研究科修士課程米川賢治氏、その他多くの友人に深く感謝し、謝辞と致します。

2012年2月14日

唐津豊

参考文献

- [1] Tanzeem Choudhury and Alex Pentland. The sociometer: A wearable device for understanding human networks, 2002.
- [2] Emiliano Miluzzo, Cory T. Cornelius, Ashwin Ramaswamy, Tanzeem Choudhury, Zhigang Liu, and Andrew T. Campbell. Darwin phones: the evolution of sensing and inference on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pp. 5–20, New York, NY, USA, 2010. ACM.
- [3] bump technologies. bump. <http://bu.mp/>.
- [4] Apple Inc. Air drop. <http://www.apple.com/jp/macosex/whats-new/features.html#airdrop>.
- [5] Apple Inc. Siri. <http://www.apple.com/iphone/features/siri.html>.
- [6] Google Inc. Google mobile app. <http://www.google.com/intl/ja-ALL/mobile/google-mobile-app/>.
- [7] 大坊郁夫. 対人コミュニケーションの社会性. 対人社会心理学研究, No. 1, pp. 1–16, 2001.
- [8] Alex Bavelas. Communication patterns in task - oriented groups. *Journal of the Acoustical Society of America*, Vol. 22, pp. 725–730, 1950.
- [9] 神戸 雅一・山本 修一郎. 会話の場に対する構造モデルの提案. 人工知能学会 第六回知識流通ネットワーク研究会, 2009.
- [10] 柿本敏克. 集団間関係研究のコミュニケーション論的位置づけ. 群馬大学社会情報学部研究論集, Vol. 8, pp. 79–91, 2001.
- [11] Trevor Pering, Roy Want, Barbara Rosario, Shivani Sud, and Kent Lyons. Enabling pervasive collaboration with platform composition. In *Proceedings of the 7th International Conference on Pervasive Computing*, Pervasive '09, pp. 184–201, Berlin, Heidelberg, 2009. Springer-Verlag.
- [12] 別所克人, 松永昭一, 大附克年, 廣嶋伸章, 奥雅博, 林良彦. 話題構造抽出に基づく会議音声インデクシングシステム (音声, 聴覚). 電子情報通信学会論文誌. D, 情報・システム, Vol. 91, No. 9, pp. 2256–2267, 2008-09-01.
- [13] R. Venkatesha Prasad, Abhijeet Sangwan, H. S. Jamadagni, Chiranth M.C, Rahul Sah, and Vishal Gaurav. Comparison of voice activity detection algorithms for voip. In *Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, ISCC '02, pp. 530–, Washington, DC, USA, 2002. IEEE Computer Society.
- [14] 傳田遊亀, 田中貴雅, 中山雅人, 西浦敬信, 山下洋一. 話者方位情報とゼロ交差情報に基づくハンズフリー発話区間検出の評価 (セッション 5 : 音声認識 + 音声合成). 情報処理学会研究報告. SLP, 音声言語情報処理, Vol. 2007, No. 11, pp. 109–114, 2007-02-09.

- [15] D.A. Reynolds and R.C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, Vol. 3, No. 1, pp. 72–83, jan 1995.
- [16] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, Vol. 10, No. 1-3, pp. 19 – 41, 2000.
- [17] Francesco Calabrese, Francisco Pereira, Giusy Di Lorenzo, Liang Liu, and Carlo Ratti. The geography of taste: Analyzing cell-phone mobility and social events. In *Pervasive Computing*, Vol. 6030 of *Lecture Notes in Computer Science*, pp. 22–37. Springer Berlin / Heidelberg, 2010.
- [18] Alvin Chin, Hao Wang, Lijun Zhu, Bin Xu, and Hao Wang. Connecting people through physical resources in an office environment. In *Proceedings of the 13th international conference on Ubiquitous computing*, UbiComp '11, pp. 475–476, New York, NY, USA, 2011. ACM.
- [19] 森脇紀彦, 佐藤信夫, 脇坂義博, 辻聡美, 大久保教夫, 矢野和男. 組織活動可視化システム「ビジネス顕微鏡」(対面コミュニケーション-顔を中心的メディアとした). 電子情報通信学会技術研究報告. HCS, ヒューマンコミュニケーション基礎, Vol. 107, No. 241, pp. 31–36, 2007-09-23.
- [20] Trinh-Minh-Tri Do and Daniel Gatica-Perez. Groupus: Smartphone proximity data and human interaction type mining. In *15th annual International Symposium on Wearable Computers*, June 2011.
- [21] 岡本昌之, 池谷直紀, 西村圭亮, 菊池匡晃, 長健太, 服部正典, 坪井創吾, 芦川平. 端末音声の相互相関に基づくアドホック会話の検出. 日本データベース学会論文誌, Vol. 7, No. 1, pp. 163–168, 2008-06-27.
- [22] HTC. Htc desire hd. <http://www.htc.com/jp/smartphones/htc-desire-hd/>.
- [23] CyanogenMod.com. Cyanogenmod. <http://www.cyanogenmod.com/>.
- [24] google.com. Android developers. <http://developer.android.com/sdk/android-2.3.4.html>.
- [25] Mitsuyoshi Nagao, Hidenori Kawamura, Masahito Yamamoto, and Azuma Ohuchi. Acquisition of personal tourism activity information based on gps log mining method. *Proceedings of the Fifth Asia-Pacific Industrial Engineering and Management Systems Conference*, 2004.

付録 A

ユーザスタディ評価用紙

以下に本研究 2 章で述べた、日常生活における会話分類調査で被験者に配布した実験資料とアンケート用紙を示す。

Date: _____
* 会話に参加していた人数は何人ですか？ _____人
* 会話を続けていた時間を選択してください 10分以内・～30分・1時間・3時間・3時間以上
* 事前にスケジュールリングされていた会話ですか？ 1・2・3・4 <small>突発的に生じた会話である</small> <small>事前に予定されていた会話である</small>
* 会話の重要度を4段階で採点してください 1・2・3・4 <small>重要でない</small> <small>重要である</small>
* 会話の内容や目的を一言で記述してください _____
* 会話中に使用した情報アプリケーションがあれば記入してください _____

図 A.1 アンケート

実験の手引き

人々が日常生活を送る中で、他者と Face-to-Face で行う会話についての傾向を探るため、アンケート調査にご協力ください。本実験で扱う「会話」とは、「人々が音声を発してコミュニケーションを取る行為」を差し、本実験では会話時に形成される人々のグループの形成傾向について調査を行います。なお、本実験で収集した情報は研究目的でのみ利用させていただきます。

* 実験目的

日常生活内で行われる Face-to-Face の会話の傾向の調査。

* 実験期間

2011/7/6 ~ 2011/7/12 (1週間)

* 実験手順

1. 配布したペーパーアンケートシートと筆記用具を実験期間中、常に持ち歩いてください。
2. 誰かと Face-to-Face で会話を行った時、その都度、アンケートシートを1枚記入してください。(記入例は以下を参照)
3. 実験期間終了後、記入したアンケートシートの束を実験責任者に提出してください。

* 本実験における会話

本実験中で扱う「会話」とは「人々が音声を発することで行う実世界上でのコミュニケーション」を差します。そのため、友人と行う雑談やオフィスの廊下で上司と行う立ち話といった会話はもちろん、大学で出席する講義、著名人の講演会、研究室やサークルのミーティング、日取りを決めたディスカッションといった会話も1つの会話として扱うことに注意してください。

* アンケート記入例

Date:

会話が始まった日時を記入してください。

例 2011/07/05/19:30

会話への参加人数:

1つの会話へ参加している話し手と聞き手の合計人数を記入してください。例えば友人との雑談であれば雑談に参加している人の数を記入します。大学での講義であれば教室内で先生の話を聞いている人と話をしている先生を合わせた大体の合計人数です。ミーティング、ディスカ

セッションであれば、それらの会話で討論している人と議論を聞いている人の総人数を差します。
会話が事前にスケジュールリングされていたかどうか

会話が始まった時の参加者の間で、その会話を行う日取りの取り決めがされていれば、高い値を選択してください。逆に突発的に会話は始まった場合は低い値を選択してください。例えば、大学の授業や定例ミーティングなど事前に参加者が集まる日取りが決まっているような会話、定期的集まる会話であれば、高いスコアを選択してください。逆に立ち話や雑談など、特に参加者が集まる、会話を行う予定を事前に共有されていないような会話であれば、低いスコアを選択してください。

会話の重要度を選択してください

会話を行った内容や得られた情報が重要だと感じた場合は高いスコアを選択してください。例えば会話をした内容を後でまとめたり、思い返したりする必要があるなどと感じた場合はスコアが高くなります。

会話の内容や目的の記入

会話内容や会話目的を一言で記述してください。以下は記入例となります。

- ・ オブジェクト指向プログラミングの授業
- ・ ○○先輩との研究ネタ相談
- ・ 研究室新人ミーティング
- ・ 昼ご飯に関する雑談
- ・ ○○先生と廊下で立ち話

会話中で使用したアプリケーション

会話を行っている最中に利用したアプリケーションや情報サービスがあれば自由に記入してください。

以上です。

次ページの被験者情報シートへの記入をお願いします。

被験者情報シート

A. 氏名

B. 年齢

C. 性別

1. 男 2. 女 3. その他

D. 所持している携帯デバイス

図 A.4 被験者情報シート

付録 B

ユーザビリティ評価アンケート

以下にアプリケーションユーザビリティ評価に使用したアンケート用紙を示す。

Milvus 被験者実験評価用紙

以下の回答項目を全て記入して下さい

1. 氏名

2. 年齢

3. 性別

男性 ・ 女性

4. サービス対象の指定について最も負荷が低かった

- A. 電子メール / B. インタラクション型共有サービス (Bump)
C. 本アプリケーション

5. サービス対象の指定について最も負荷が高かった

- A. 電子メール / B. インタラクション型共有サービス (Bump)
C. 本アプリケーション

6. 会話コンテキスト情報は携帯端末サービスにおいて有用である

有用でない 1 / 2 / 3 / 4 有用である

7. 会話コンテキスト情報を実際に利用するコンテキストウェアシステムを利用したい

利用したくない 1 / 2 / 3 / 4 利用したい

以上です ご協力ありがとうございました

図 B.1 ユーザビリティ評価アンケート