修士論文

マイクロブログからの典型的使用場面付き辞書の構築

岡 利成

主指導教員 白井 清昭

北陸先端科学技術大学院大学
先端科学技術研究科
（情報科学）

令和２年３月

Abstract

A lexicon is a database of words, which contains information of words such as pronunciation, part of speech, and synonym. It is one of the fundamental knowledge for natural language processing. In particular, it is very useful for various researches to construct a lexicon that compiles words with time or place where each word is frequently used or with the information of people who frequently use each word. The goal of this research is to automatically construct such a lexicon, i.e. a lexicon with typical situations of words. This study considers three types of typical situations of words: (1) time and (2) place where a word is frequently used as well as (3) job of a person who frequently uses a word. For example, a typical situation of time of the word "breakfast" is morning. In this study, categories of typical situations are defined as follows: [morning], [noon], [evening], [night], and [midnight] for time, 47 prefectures for place, and 44 representative jobs such as [doctor] and [teacher]. From Twitter, one of the microblogs, we collect texts with information of time, place and job, and identify typical situations where the words frequently appear in the text.

Although several previous studies aimed at identifying typical situations of words and applied them to specific applications of natural language processing, no attempt has been made to construct a lexicon with typical situations of words that can be widely used in general. In addition, time or place was considered as a typical situation in the previous work, but a job has not been considered as a typical situation. This study is the first attempt to consider the job as the typical situations of words. Furthermore, although typical situations of nouns were considered so far, we also consider other types of words, that is, verbs, adjectives, adverbs, and hashtags, as words to be compiled in a lexicon.

In the proposed method, a lexicon with typical situations of words is constructed by the following procedures. First, tweets annotated with a category of time, place, and job are collected using TwitterAPI Tweepy. As for the place, tweets are searched using the place name code of a prefecture as a query, then tweets annotated with the category of the prefecture are retrieved. As for a job, we collect tweets posted by job users. A "job user" is a Twitter user who has a certain occupation (job), which is automatically collected by our proposed method described later. As for the time, since a time stamp is attached to all tweets as metadata in Twitter, we reuse a collection of tweets with place categories as tweet data with time categories. Next, after several preprocessings, word segmentation and part of speech tagging are performed on tweets, then nouns, verbs, adjectives, adverbs, and hashtags are extracted as candidates of words to be compiled in a lexicon with typical situations.

Typical situations of these candidates of words (time, place, or job that are

highly associated with a word) are identified by the following three methods. The first is a method using Pointwise Mutual Information (PMI). Correlation (co-occurrence) between words and categories is measured by PMI to identify typical situations of words. The second method is based on Kleinberg's burst detection algorithm. Regarding a sequence of categories as a virtual time series, we identify the time period (corresponding to the category) where the frequency of a word is sharply increased by Kleinberg's method. The detected category is set as the typical situation of the word. The frequency of words in each category is measured by the number of tweets containing a word. Hereafter, we call this method "Kleinberg-tweet method". The third method is also based on Kleinberg's burst detection algorithm, but the frequency of words in each category is measured by the number of users who use the word. Hereafter, we call this method "Kleinberg-user method". Finally, when the score calculated by each method is greater than a predefined threshold, the category is specified as a typical situation of the word, then the words and their identified typical situations are added to the lexicon.

In the above method, a list of job users is required to obtain tweets annotated with jobs. It is obtained semi-automatically as follows. First, for a given job, user profiles in Twitter account are searched using the job name as a query. Then, a user who actually has the job (we call it as a seed job user) is manually selected. Next, the followees of the seed job user is retrieved. When the profile of the followee includes the job name, it is added to the list as a new job user. This procedure is repeated recursively until 100 job users are obtained.

Several experiments are conducted to evaluate the proposed method. First, we evaluated the method of identifying typical situations of words using PMI. For the place category, we selected several pairs of a word and its identified typical situation (place category) and manually evaluated whether they were appropriate. However, the accuracy was insufficient, i.e. 0.26. Next, we evaluated the Kleinberg-tweet method. The top 20 words of the score calculated by the Kleinberg-tweet method for each category were manually evaluated whether they are appropriate by two subjects. Then the accuracy, the ratio of the words that are judged as appropriate to the total number of words, is measured. The accuracy was 0.44, 0.79, and 0.52 for the time, place, and job category, respectively. On the other hand, the κ coefficients of judgement of two subjects were 0.76, 0.77, and 0.53 for time, place, and job, respectively. Similarly, the Kleinberg-user method was evaluated. The accuracy was 0.59, 0.90, and 0.92 for the time, place, and job category, respectively, which outperformed the Kleinberg-tweet method. The κ coefficients of judgement of two subjects were 0.83, 0.41, and 0.49 for time, place, and job, respectively. In the Kleinberg-tweet method, when one user use the same word many times, the number of tweets of that word increases, and the word is wrongly detected

as it is highly related to a specific category. In contrast, since the Kleinberg-user method detects words that are used by many users, such errors are suppressed. Finally, using the Kleinberg-user method, we constructed a lexicon consisting of 1,152 words with time typical situations, 6,793 words with place typical situations, and 199,475 words with job typical situations.

概要

辞書とは，単語の読み，品詞，類義語などの情報が記載された単語のデータベースであり，自然言語処理に欠かせない知識である．中でも単語が頻繁に使用される時間や場所，またその単語をよく使用する人物の情報が付与された辞書は，様々な場面で利用できるため，利用価値が高い．本研究は，単語の典型的使用場面の情報を持つ辞書を自動的に構築することを目的とする．本研究における単語の典型的使用場面とは，その単語がよく使われる時間，場所，ならびにその単語をよく使う人の職業の 3 種とする．例えば,「おはよう」の時間の典型的使用場面は朝である．典型的使用場面のカテゴリの定義は，時間については【朝】【昼】【夕方】【夜】【深夜】，場所については 47 都道府県，職業については【医者】【教師】などの代表的な 44 種の職業とする．マイクロブログの一つである Twitter から，時間・場所・職業の情報と共にテキストを収集し，テキストで出現する個々の単語について，それが高頻度で使用される場面を特定する．

単語の典型的使用場面を特定し，自然言語処理に応用した先行研究はいくつかあるが，汎用的な辞書として整備する試みはこれまで行われていなかった．また，典型的使用場面として考慮されていたのは時間もしくは場所であり，職業の典型的使用場面を特定する試みは行われていない．さらに，先行研究の多くは名詞を対象としているが，本研究は動詞，形容詞，副詞，ハッシュタグも辞書に登録するべき単語の候補とする点が異なる．

提案手法では以下の手続きによって単語の典型的使用場面付き辞書を構築する．まず，時間，場所，職業のカテゴリが付与されたツイートを TwitterAPI の Tweepy を用いて収集する．場所については，都道府県を表わす地名コードをクエリとしてツイートを検索し，都道府県のメタデータが付与されたツイートを収集する．職業については，後述するアルゴリズムで収集された職業ユーザ (実際にその職についているユーザ) が投稿しているツイートを収集する．時間については，全てのツイートには投稿時間がメタデータとして付与されているため，場所カテゴリ付きのツイートを集合を時間カテゴリ付きのツイート集合として流用する．次に，ツイートを形態素解析し，名詞，動詞，形容詞，副詞，ハッシュタグを抽出し，典型的使用場面付き辞書の候補単語とする．

これらの候補単語に対して，3 つの手法で典型的使用場面，すなわちその単語と関連の深い時間，場所，職業のカテゴリを特定する．1 つ目は自己相互情報量 (PointwiseMutualInforation:PMI) を用いる手法である．単語とカテゴリの相関の強さ (共起の強さ) を測り，単語の典型的使用場面を特定する．2 つ目は Kleinberg のバースト検知アルゴリズムに基づいた手法である．カテゴリの列を仮想的な時系列とみなし，特定の時間帯に単語の使用頻度が急激に増加することを検出し，検出されたカテゴリをその単語の典型的使用場面とする．それぞれのカテゴリにおける単語の使用頻度は，候補単語を含むツイートの数とする．以下，これを Kleinberg-tweet 手法と呼ぶ．3 つ目は，それぞれのカテゴリにおける単語の使用頻度をその単語を使用したユーザ数で算出し，Kleinberg のバースト検知アルゴリズムを適用

する手法である．以下，これを Kleinberg-user 手法と呼ぶ．最後に，それぞれの手法で算出されたスコアが閾値以上であるとき，カテゴリを単語の典型的使用場面として特定し，その単語とカテゴリの組を辞書に登録する．

上記の手法で職業のメタデータが付与されたツイートを得るためには，職業ユーザのリストが必要である．本研究ではこれを半自動的に獲得する．まず，職業名がプロフィールに記入されているユーザを検索し，職業が明らかなユーザ (親ユーザと呼ぶ) を人手で選択する．親ユーザがフォローし，かつ職業名がプロフィール欄に記載されたユーザの職業は，親ユーザの職業と同じとみなし，これを職業ユーザとみなす．この操作を 100 名の職業ユーザが得られるまで再帰的に繰り返す．

提案手法の評価実験を行った．まず，PMI によって単語の典型的使用場面を特定する手法を評価した．場所カテゴリについて，単語と典型的使用場面として特定されたカテゴリの組をいくつか選択し，適切であるかを人手で評価した．しかし，その正解率は 0.26 となり，十分な結果が得られなかった．次に，Kleinberg-tweet 手法を評価した．各カテゴリから Kleinberg-tweet 手法で計算されたスコアの上位 20 単語を作業者 2 名によって人手で評価した．正解率は，時間カテゴリは 0.44，場所カテゴリは 0.79，職業カテゴリは 0.52 であった．同様に Kleinberg-user 手法を評価した．正解率は，時間カテゴリは 0.59，場所カテゴリは 0.90，職業カテゴリは 0.92 となり，Kleinberg-tweet 手法を上回った．これは，Kleinberg-tweet 手法では，一人のユーザが同じ単語を繰り返し使うとき，その単語のツイートの数が多くなり，特定のカテゴリとの関連が深い単語として誤って検出されるのに対し，Kleinberg-user 手法では多くのユーザが使う単語が検出されるため，このような誤検出が少なくなるためであった．最終的に，Kleinberg-user 手法を用いて，時間カテゴリが付与された単語を 1,152 個，場所カテゴリが付与された単語を 6,793 個，職業カテゴリが付与された単語を 199,475 個を含む典型的使用場面付き辞書が構築された．

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

辞書とは，単語の読み，品詞，類義語などの情報が記載された単語のデータベースであり，自然言語処理に欠かせない知識である[1]．辞書には様々なものがあるが，単語が頻繁に使用される時間や場所，またその単語をよく使用する人物の情報が付与された辞書は，様々な場面で利用できる．このような辞書が有効的に使用されている例を以下に示す．

時間 社会学の分野では，社会の流行を予測することは，CM 制作，ドラマ制作などにおいて非常に重要である．そのため，自由国民社が 1 年毎に発行する現代用語の基礎知識 [6, 7] を活用して，流行分野を予測する手法が提案されている [3]．また，医療分野では，インフルエンザの流行時期の早期発見が求められている．そのため，テキストが頻繁に投稿される Twitter からタイムスタンプとともにツイートを収集し、これを時間付きのテキストとして活用して，インフルエンザの流行時期を予測する研究が行われている [1]．これらの研究例では時間の情報が付与されたテキストを必要とする．現代用語の基礎知識やツイートには時間の情報が付与されているが，そうでないテキストも存在する．一方，単語とそれに関連の深い時間が記載された辞書があれば，テキストが書かれた時間を推測することも可能であり，時間の情報が付与されていないテキストを社会のトレンドの予測やインフルエンザの流行予測に活用することもできる．

場所 観光情報学の分野では，知らない土地に旅行するときに，その土地の適切な土産物情報を収集するニーズがある．そのため，Q & A サイトから土産物情報を抽出し，ユーザに提示するアプリケーションが開発されている [8]．また，自然言語処理の分野では，人間らしい会話を実現する非タスク指向型対話システムが盛んに研究されている．例えば，ユーザの発話に含まれる単語から場所の状況を推定し，その場所の状況に応じて人間らしい応答を返す手法が考案されている [2]．

[1]「辞書」とは，一般には国語辞典のような単語の意味を定義した書籍を表わすが，本論文ではこのような自然言語処理用知識を指す．

人物 マーケティングの分野では，テレビ番組や商品などの評価対象に対する口コミがポジティブであるかネガティブであるかを判定し，評価対象に対する評判を推定する評判情報分析が注目されている．この際，ユーザを性別・年齢・職業に分け，それぞれのカテゴリ毎に評価対象に対する評判を分析することは有用である．そのため，ユーザが投稿したテキスト (レビュー) から，性別・年齢・職業といったプロフィールの情報を推定する技術が研究されている [4]．このとき，特定の性別，年齢，職業の人がよく使う単語の辞書があれば，ユーザプロファイルの自動推定に有用であると考えられる．また，アンケート調査では，被験者の自由回答のテキストを読んで，それを被験者の職業に分類することが求められる．これを人手で行うことは，そのコストが高いこと，分類に専門的知識を持つ人を必要することなどの理由で困難である．そのため，自由回答テキストに出現する単語を素性として被験者の職業を推定する研究が行われている [17]．このとき，職業に関連の深い単語の辞書があれば，職業を推定する分類器を学習する際の有効な素性になりうる．

しかしながら，上記のような個々の研究・システムにおいては，単語が典型的に使われる場面 (時間，場所，人物など) が推定されることがあるが，独立した知識データベースとして，つまり典型的な使用場面の情報が付与された辞書として整備された試みはこれまで行われていない．典型的な使用場面の情報が付与された辞書を整備することは，自然言語処理の研究分野ならびに自然言語処理とその他の分野との融合分野の研究の支えとなり，様々な応用システムの研究開発を促進する可能性がある．

1.2 目的

本研究は単語の典型的使用場面の情報を持つ辞書を自動的に構築することを目的とする．本研究における典型的使用場面とは，単語が頻繁に使用される時間・場所・人物の 3 種類と定義する．以下にその詳細を述べる．

時間 単語が特定の時間によく使われるとき，その時間を単語の典型的使用場面とする．時間カテゴリとして,【朝】【昼】【夕方】【夜】【深夜】の 5 種類を定義する．例えば,「おはよう」の時間の典型的使用場面は【朝】である．

場所 単語が特定の場所でよく使われるとき，その場所を単語の典型的使用場面とする．場所カテゴリは 47 都道府県と定義する．例えば,「はいさい」の場所の典型的使用場面は【沖縄県】である．

人物 人物のプロフィールには，性別，年齢，出身地など様々なものがあるが，本研究では職業を対象とする．すなわち，単語が特定の職業の人によってよく使われるとき，その職業を単語の典型的使用場面として定義する．職業カテ

ゴリの詳細は 3.2.3 項で述べる．例えば,「注射」の職業の典型的使用場面は【医師】である．

マイクロブログから時間・場所・職業の情報と共にテキストを収集し，各単語について，それが高頻度で使用される場面を特定する．本研究において，テキスト収集の対象とするマイクロブログは Twitter である．そして，単語と特定した典型的使用場面を結びつけて，自然言語処理システムで利用できるように辞書として整備する．さらに，辞書の利便性を高めるために，典型的使用場面で使われる傾向の強さを表すスコアを算出し，辞書に登録する．構築した辞書は研究者が広く利用できるよう公開する．

1.3 本論文の構成

本論文の構成は以下の通りである．２章では，関連研究を紹介するとともに，本研究の提案手法の妥当性や，関連研究と本研究の違いについて述べる．３章では，提案手法について述べる．Twitter からツイートを収集する手法，単語の典型的使用場面を推定する手法などについて説明する．４章では，提案手法によって辞書を構築し，その品質を評価する実験について述べる．最後に５章では，本論文のまとめと今後の課題を述べる．

第2章　関連研究

本章では関連研究について述べる．2.1 節では，単語の典型的使用場面に関する研究を紹介する．2.2 節では，提案手法は Twitter ユーザの職業を推定する処理を含むため，ツイートからユーザの (職業などの) プロフィールを推定する過去の研究について述べる．2.3 節では，本研究では Twitter を辞書構築の知識源として利用するため，Twitter を対象とした研究の概要を紹介する．最後に，2.4 節では，先行研究と本研究の違いを論じる．

2.1　単語の典型的使用場面に関する研究

典型的使用場面の情報が付与された辞書やコーパスを構築する研究がいくつか行われている．2.1.1 項では時間の典型的使用場面について，2.1.2 項では場所について，関連研究を紹介する．

2.1.1　時間の典型的使用場面に関する研究

情報検索や情報抽出の分野では，テキストの時間表現に対して実時間のタグを付与することが求められることがある．Pustejovsky らは，テキスト中の時間表現を適切に取り扱うために，時間表現の種類として日付表現（「一九二九年二月」など），時刻表現（「午前十時ごろ」など），時間表現（「その間」など），頻度集合表現（「毎日」など）を定義し，またそのアノテーション基準を定めた TimeML を提案している [14]．さらに，Pustejovsky らは，TimeML の定義を基に，新聞記事に時間表現タグをアノテーションしたコーパスである TimeBank を人手により構築している [15]．

保田らは，TimeML の定義を基に，現代日本語書き言語均衡コーパス[1]に対して時間表現のタグを付与した [18]．しかし，日本語については，テキストに対して時間情報をアノテーションする試みはそれほど多くはない．

自由国民社が毎年に発行している「現代用語の基礎知識」は，現代人として知っておく必要のある用語や，マスコミ・ウェブなどで使用されるその年の新語や流行語が記載されている [6, 7]．この書籍は，国語辞典のように単語の意味が定義されている．また，1 年毎に編集者の手によって単語の追加・加筆・削除が行われて

[1] https://pj.ninjal.ac.jp/corpus_center/bccwj/，閲覧日 2020/01/19.

いる.「現代用語の基礎知識」は，単語に年の時間情報が付与された辞書の一種と言える．この場合，書籍の発行年が時間情報 (時間のメタデータ) となる．

2.1.2 場所の典型的使用場面に関する研究

空間情報学の分野では，ウェブ上のニュースやブログなどのテキストに対し，テキスト内で触れられているイベントが発生した場所を地図上に表示する研究が行われている．GeoNLP プロジェクト[2]は，自然言語文に記載される地名や施設といった場所を表わす単語を抽出し，その単語に位置情報を付与するジオタギングシステムを構築し，これを多くのユーザで共有するプラットフォームを提供している．主に GeoNLP ソフトウェア (テキストから地名を抽出するソフトウェア)，GeoNLP データ（GeoNLP 地名辞書のアップロード・ダウンロードが可能なウェブサイト），GeoNLP サービス（地名キーワードや地名を含む文章をクエリとして，その文章で言及されている場所を地図上に表示する簡易的なウェブサービス）の 3 つの要素から構成されている．GeoNLP データにおいては，一般ユーザが地名や施設などの単語に位置情報をタグ付けした GeoNLP 地名語辞書を，その情報源やデータ加工の方法と共にアップロードしている．しかし，概要覧に詳細な辞書の構築手順を記載しているユーザは少なく，人手もしくは自動的に構築されたのかが不明な辞書が存在する．

松田らは，災害の際に SNS(Social Networking Service) に投稿される情報が災害時の救助の手助けとなることを指摘し，SNS 上の情報を救助に活用するためには「いつ」「どこで」についての情報を特定することが必要であると述べている [10]. そのため，ツイートに出現する地名や施設などの単語に対して，その緯度・経度の位置情報を人手でアノテーションしたコーパスを構築している．

観光などで知らない土地に旅行するとき，その土地の著名な土産物を知らないことがある．川野らは，地域毎にその土地の土産物情報を提供するアプリケーションを開発している [8]．Q & A サイトのお土産カテゴリから，テキスト内に都道府県を表現する単語が存在する場合に，同テキスト内に共起する単語を土産物を表わす単語として検出している．まず，三重県のお土産を対象とし，Q & A サイトのテキストを形態素解析によって単語に分割し，単語 N-gram を取得した．6,751 個の単語 N-gram を人手で確認し，69 個の N-gram が土産物名と判定された．次に，6,751 個の単語 N-gram に対して残差 IDF を計算し，スコア順にソートし，その分布を調べた．その結果，人手で土産物名と判定された単語 N-gram の半数が，残差 IDF のスコア上位 10%に分布したと報告している．

対話システムでは，人間らしい雑談を目的とした非タスク指向型対話システムが求められている．服部は，システムと人間が時間と空間の情報を共有することでより人間らしい雑談ができるとし，このような非タスク指向型対話システムを

[2]https://geonlp.ex.nii.ac.jp/page/about/，閲覧日 2020/01/19.

実現する方法を提案している [2]．地名や地域名を検索クエリとしてウェブ検索を行い，ある場所に関連するウェブページを取得し，その場所でよく使われる単語からユーザへの応答を生成する．論文では，対話が行われている場所が「フランス」のとき，人間の「こんにちは」という発話に対して対話システムが「ボンジュール」と返答する例が挙げられている．

2.2 ユーザのプロフィールの自動推定に関する研究

マイクロブログでは，一般にユーザのプロフィールの入力は任意であり，プロフィールが必ずしも記入されているとは限らない．しかし，ユーザの職業，性別，年齢などのプロフィール情報はマーケティングの分野などで必要とされている．したがって，ユーザが投稿したツイートからそのユーザのプロフィールを推定する様々な研究が行われている．

奥谷と山名は，ツイートに含まれるメンション情報からユーザの交友関係を判断し，ユーザの周囲のコミュニティから属性タグを抽出することで，ユーザのプロフィールを推定する手法を提案している [12]．評価実験では，プロフィールの推定精度が 58.6%と報告している．

Preotiuc-Pietro らは，Twitter ユーザの職業を推定する手法を提案している [13]．まず，Twitter のプロフィールを参照し，9 種類の職業について，それを職業とするユーザを収集する．これを訓練データとして，そのユーザが投稿したツイートのテキストやユーザのフォロー・フォロワー関係などを素性とした教師あり機械学習により職業を分類するモデルを学習する．評価実験では，職業分類の正解率が 52.7%と報告している．

2.3 Twitter を対象とした研究

近年，マイクロブログの普及により一般の人々が情報を発信することが盛んになっている．このような情報を有効活用する試みは多数行われており，自然言語処理の分野では，リアルタイム性・膨大なデータ量・豊富な API などの点で，テキストマイニングによる知識獲得の情報源として注目されている．マイクロブログを対象としたテキストマイニングはもはや研究者の中では広く知られており，マイクロブログマイニングとも呼ばれている．現在研究されているマイクロブログを対象とした研究の例を以下に示す [11]．

- 分析技術
- Authority
- 評判分析

- 実世界の動向の予測
- マイクロブログの書き手の属性推定
- マイクロブログのトピック同定
- トレンド分析
- 自動要約
- 情報の信頼性評価
- スパムフィルタリング

Twitter はマイクロブログのひとつである．Twitter は，ユーザに対し，ツイートと呼ばれる短文のテキストを投稿するプラットフォームを提供する．ツイートには投稿時刻・投稿者・メンション情報（リプライに関する情報）などのメタデータが付与されている．また，フォロー，リツイート，リプライなど，ユーザ間のつながりを促す機構も用意されている．このようなユーザ間のつながりの集合はソーシャルグラフと呼ばれ，これを利用することで，テキストだけを対象とした単純なテキストマイニングでは発見できない知見が見出されることもある．

一方，Twitter は一般に SNS の一種とも位置付けられる．石井は，複数の SNS，具体的には Facebook，mixi，モバゲータウン，グリー，Twitter を比較している [5]．個人情報の開示の有無について注目し，匿名的な SNS はユーザの利用頻度が多い傾向があることを報告している．特に Twitter は他の SNS よりも突出している．本研究の目的は大量のテキストから単語の典型的使用場面付きの辞書を構築することであるが，Twitter は，ユーザの使用頻度が高く日々膨大なテキストが生成されている点，様々なユーザによって投稿されたテキストが入手できる点，投稿時間などのメタデータを利用できる点，ユーザ間のソーシャルグラフの情報を利用できる点など，辞書構築の知識源として適した性質を持っている．

Twitter は，実世界を観測するためのソーシャルメディアとみなすこともできる．様々な研究で，Twitter のデータを分析することで，実世界でどのようなイベントが発生しているかを推定しているためである．榊と松尾は，実世界の情報を直接計測する物理センサに対して，Twitter ユーザをソーシャルセンサとして捉えている [16]．ソーシャルセンサは物理センサの代わりとなるかを調査したところ，物理センサでは観測できない情報を取得できることや，手法を工夫することでソーシャルセンサの信頼性や安定性を保つこともできると述べている．

2.4 本研究の特色

本節では，本研究の特徴や新規性について述べる．関連研究との違いを 6 つの視点から考察し，本研究の特色を明らかにする．

1 つ目の視点は，典型的使用場面の種類の違いである．関連研究 [6, 7, 15, 18, 10] や GeoNLP では，単語の典型的使用場面は時間もしくは場所であった．本研究で考慮する典型的使用場面は，時間・場所に加え，職業も含まれる点が異なる．職業の情報が付与された辞書の構築は初めての試みである．

2 つ目の視点は，単語の典型的使用場面を自動的に推定するか人手で推定するかの違いである．関連研究 [6, 7, 15, 18, 10] では，典型的使用場面の情報は基本的には人手で付与されている．これに対し，本研究では単語の典型的使用場面を自動的に特定する．これにより，多くの単語に対してその典型的使用場面を決定し，典型的使用場面の情報が付与された大規模な辞書を構築できる．

3 つ目の視点は，場所を特定する手法の違いである．関連研究 [8, 2] では，テキストが書かれた場所は，テキスト内の単語などから推測していた．本研究では，ツイートが投稿された場所を特定する際には，ツイートのメタデータを用いて正確に特定する．

4 つ目の視点は，典型的使用場面の時間の粒度の違いである．関連研究 [6, 7] では，時間の粒度 (単位) は年である．本研究では，ツイートのタイムスタンプを利用するため，辞書に登録する時間カテゴリの粒度を柔軟に変更できる．本論文では，時間カテゴリは【朝】【昼】【夕方】【夜】【深夜】の 5 つであり，年よりも粒度が細かい．また，時間カテゴリを【1 時】【2 時】などのように時間単位でより細かく定義したり，曜日，月，季節などより粗く設定したりすることも容易に可能である．

5 つ目の視点は，ユーザの職業を推定する手法の違いである．Twitter では，ユーザの職業はメタデータとして付与されていないため，本研究では，ある単語をよく使う人の職業を自動的に推定する．人手により特定の職業を持つユーザを収集することも考えられるが，大規模な辞書を構築するためには，人手による作業をできるだけ省く必要があり，ユーザの職業も自動推定する方が望ましい．関連研究 [12, 13] では，テキストや Twitter のソーシャルグラフの情報を使用し，教師あり機械学習で職業を推定している．しかし，職業を分類する実験では正解率がそれほど高くない．また，正解の職業が付与された Twitter ユーザの集合を人手で構築するコストが高く，機械学習のための大規模な訓練データを確保することが難しいという問題点もある．本研究では，教師あり機械学習で職業を自動推定するのではなく，まず人手で職業毎に数人のユーザを特定し，その人によってフォローされているユーザを同じ職業の人物として特定する手法を提案する．この手法は大規模な職業付きユーザのデータを必要としない利点がある．

6 つ目の視点は，辞書に登録する単語の種類の違いである．関連研究 [6, 7, 15, 18, 10] や GeoNLP では，典型的使用場面を特定する対象となる単語は，地名・時間などの名詞のみであった．本研究では，時間，場所，職業のいずれにおいても，辞書に登録する単語は自立語 (名詞・動詞・形容詞・副詞) とハッシュタグを対象とする．名詞以外の単語やハッシュタグについても典型的使用場面の情報を付与する点に本研究の特色がある．1.1 節で述べたような応用システムに典型的使用場

面付きの辞書を使うときは，名詞だけでなく様々な品詞の単語が収録されている方が望ましい．

表 2.1 は上記の議論を簡潔にまとめたものである．

本研究は，様々な単語の典型的使用場面の情報を自動的に特定し，特定の自然言語処理システムを想定しない汎用的な辞書として整備する初めての試みである．典型的使用場面情報付き辞書は様々な自然言語処理応用システムで利用可能であり，これを構築する意義は大きい．

表 2.1: 本研究と関連研究の違い

	本研究	先行研究
典型的使用場面の種類	場所・時間・職業	場所・時間 [6, 7, 15, 18, 10]・GeoNLP
典型的使用場面の推定手法	自動	人手 [6, 7, 15, 18, 10]
場所の特定手法	ツイートのメタデータ	テキスト内の単語 [8, 2]
時間カテゴリの粒度	柔軟に変更可能	年 [6, 7]
職業ユーザの特定手法	教師なし	教師あり [12, 13]
登録する単語の種類	自立語・ハッシュタグ	名詞 (地名・時間名) [6, 7, 15, 18, 10]・GeoNLP

第3章　提案手法

本章は，本研究の提案手法について述べる．提案手法は主に 4 つのステップに分けられる．その処理の流れを図 3.1 に示す．始めに,「ツイートの収集」では，典型的使用場面のカテゴリ別にツイートを収集する．次に,「候補単語の抽出」では，収集したツイートを形態素解析器により単語に分割し，辞書に登録する候補単語を抽出する．次に,「典型的使用場面の特定」では，単語と関連の深いカテゴリ (典型的使用場面) を特定し，単語が特定した場面で使用される傾向の強さを表すスコアを算出する．最後に,「辞書の構築」では，単語，その典型的使用場面のカテゴリ，スコアを一つのエントリとして辞書を構築する．

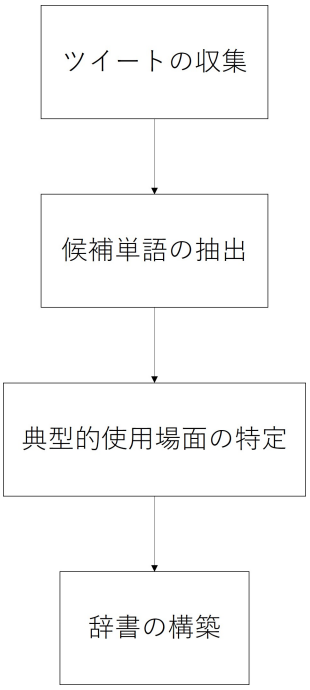

図 3.1: 提案手法の概要

3.1　典型的使用場面の定義

本研究における典型的使用場面の定義を示す．既に述べたように，本研究における単語の典型的使用場面には，時間，場所，職業の 3 種がある．

時間について，単語の典型的使用場面の定義の仕方には，曜日，月，季節 (春夏秋冬) など，様々なものが考えられる．本研究では，1 日の時間帯を【朝】【昼】【夕方】【夜】【深夜】に分け，この 5 つを時間の典型的使用場面のカテゴリとする．時間カテゴリとそれに対応する時間帯を表 3.1 に示す．

場所についても，都市，地方 (北海道地方，東北地方など)，都道府県など，単語の典型的使用場面の定義の仕方には様々なものが考えられる．本研究は，都道府県を場所のカテゴリと定義する．場所カテゴリのリストを表 3.2 に示す．

職業については，単語の典型的使用場面のカテゴリは，ウェブサイト「13 歳のハローワーク」[1]を参照して決定する．「13 歳のハローワーク」は子どもが将来の職業を決める際の参考となる情報として，様々な職業の種類とその仕事内容が記載されたウェブサイトである．同サイトに掲載されている職業の中から，代表的と思われる職業を人手で 50 個選択し，これを職業のカテゴリと定義する．選択した職業カテゴリを表 3.3 に示す．なお，3.2.3 項で後述するように，提案手法を実装する際にはこの中のいくつかの職業カテゴリを除外する．

表 3.1: 時間カテゴリの定義

カテゴリ	対応する時間帯
朝	5:00-11:00
昼	11:00-16:00
夕方	16:00-19:00
夜	19:00-24:00
深夜	0:00-5:00

表 3.2: 場所カテゴリの定義

北海道	青森県	岩手県	宮城県
秋田県	山形県	福島県	茨城県
栃木県	群馬県	埼玉県	千葉県
東京都	神奈川県	新潟県	富山県
石川県	福井県	山梨県	長野県
岐阜県	静岡県	愛知県	三重県
滋賀県	京都府	大阪府	兵庫県
奈良県	和歌山県	鳥取県	島根県
岡山県	広島県	山口県	徳島県
香川県	愛媛県	高知県	福岡県
佐賀県	長崎県	熊本県	大分県
宮崎県	鹿児島県	沖縄県	

[1]https://www.13hw.com/home/index.html

表 3.3: 職業カテゴリの定義

看護師	保育士	医師	パティシエ
マネージャー	理学療法士	薬剤師	美容師
建築家	トリマー	教師	漫画家
作家	声優	飼育員	ホテルマン
歌手	プログラマ	アナウンサー	弁護士
カメラマン	議員	システムエンジニア	司書
画家	料理人	消防士	映画監督
助産師	通訳案内士	スポーツトレーナー	モデル
学芸員	俳優	客室乗務員	税理士
管理栄養士	歯科衛生士	フリーター	バーテンダー
自衛官	気象予報士	漁師	植木職人
ブロガー	芸人	マッサージ師	書道家
学生	主婦		

3.2 ツイートの収集

本研究では，Twitter から時間，場所，職業のメタデータが付与されたツイートを収集する．その際，TwitterAPI[2]を使用する．本節では，まず本研究で使用する TwitterAPI について説明し，続いて時間，場所，職業それぞれについてツイートの収集方法を示す．

3.2.1 TwitterAPI について

TwitterAPI は，Twitter 上の情報をプログラムレベルでアクセスするための開発者向けのツールである．これにより，ツイートの投稿，ツイートの検索，ユーザの情報の取得など，一般的に使用されている Twitter の機能をプログラム上で操作することができる．TwitterAPI には REST API と Streaming API が存在する．REST API は，現在 Twitter 上に存在している過去のツイートの取得やツイートの投稿などが可能である．それに対し Streaming API は，実行中に全世界から投稿されるツイートをリアルタイムで取得することができる．本研究では，過去のツイートやユーザの情報を収集するため REST API を使用する．

Twitter 社は様々なプログラミング言語の TwitterAPI の操作用のライブラリを提供している．本研究では Python のライブラリである Tweepy[3]を使用する．

TwitterAPI（Tweepy）には，Twitter の機能を操作するためのメソッドがいくつか提供されている．しかし，メソッドは完全に自由に使用できるわけではなく，

[2]https://help.twitter.com/ja/rules-and-policies/twitter-api
[3]https://www.tweepy.org/

いくつかの制限が設けられている．本研究で使用するメソッドとその仕様を以下に述べる．

tweepy.API.search()

機能：キーワードを指定してそのキーワード含むツイートを取得する．

入力：キーワード

オプション：1 回当たりに取得可能なツイート項目数（1〜100）を指定する．

出力：ツイート

制限：180 回/15 分まで，10 日間ほど前のツイートまで収集可能

最大取得数：100 ツイート× 180 回/15 分＝ 18000 ツイート/15 分

tweepy.API.friends_ids()

機能：ユーザの ID を指定してそのユーザがフォローしているユーザの ID を取得する．

入力：ユーザの ID

オプション：なし

出力：フォローしているユーザの ID

制限：15 回/15 分まで

最大取得数：フォロー数× 15 回/15 分

tweepy.API.lookup_users()

機能：任意の数（100 項目まで）のユーザの ID を指定してそのユーザのプロフィールを取得する．

入力：ユーザの ID

オプション：なし

出力：プロフィール

制限：900 回/15 分まで

最大取得数：100 ユーザ× 900 回/15 分＝ 90000 プロフィール/15 分

tweepy.API.user_timeline()

機能：ユーザの ID を指定してそのユーザのタイムライン（ツイート）を取得する．

入力：ユーザの ID

オプション：1 回当たりに取得可能なツイート項目数（1〜200）を指定する．

出力：タイムライン

制限：900 回/15 分まで，過去 3200 ツイートまで収集可能

最大取得数：200 ツイート× 900 回/15 分＝ 180000 ツイート/15 分

3.2.2 場所のツイートの収集

以下の手続きにより，場所カテゴリ (都道府県) 毎にツイートを収集する．tweepy.API.search() に「place:〈地名コード〉」を検索キーとして，地名コードがメタデータとして付与されているツイートを収集する．ここでの地名コードとは，Foursquare[4] と Yelp[5]から提供されている都道府県の位置情報である[6]．都道府県の地名コードの一覧を付録 A の表 A.1 に示す．これにより，各都道府県で投稿されたツイートを取得できる．ここでは，地名コードに該当する都道府県をそのツイートの場所のカテゴリとする．

3.2.3 職業のツイートの収集

本研究における職業カテゴリの定義は表 3.3 の通りである．それぞれの職業について，それを職業とするユーザが発信したツイートを収集する．しかしながら，Twitter のツイートには，ユーザの職業のメタデータは直接付与されていない．そのため本研究では，ユーザの職業を半自動的に推定する手法を提案し，職業カテゴリ毎に，それを職業とするユーザのリスト (以下,「職業ユーザリスト」と呼ぶ) を作成する．次に，職業ユーザが投稿したツイートを収集し，これを職業カテゴリのメタデータが付与されたツイートとする．

まず，与えられた職業に対し，職業ユーザのリストを作成する．職業ユーザリストを作成する簡単な手法は，Twitter のプロフィールに職業名が含まれているユーザを検索することである．しかし，予備実験では，このようにして得られた職業ユーザの中には，Bot ユーザや，プロフィール内に宣伝のために職業名が使われているユーザなど，不適切なものも多かった．そのため，ユーザのフォロー関係を利用して職業ユーザを得る手法を採用する．これは，フォロー関係にあるユーザは似たプロフィールを持つ可能性が高く，職業もまた同じである可能性が高いという仮説に基づいている．

職業ユーザのリストを取得するフローチャートを図 3.2 に示す．始めに，指定された職業名がプロフィールに含まれるユーザを検索し，そのユーザが真にその職業を持つかを人手で判定し，職業を持つと判定したユーザ 1 名を親ユーザとする．次に，そのユーザのフォロー一覧を tweepy.API.friends_ids() で取得する．フォロー・

[4]https://ja.foursquare.com/

[5]https://www.yelp.com/

[6]https://help.twitter.com/ja/using-twitter/tweet-location

ユーザのプロフィールを tweepy.API.lookup_users() で取得し，指定の職業名が含まれているかをチェックする．親ユーザのフォローであり，かつプロフィールに職業名が含まるとき，そのユーザを新しい職業ユーザと認定し，職業ユーザリストに追加する．以上の操作を連鎖的に行い，職業ユーザが 100 人以上収集できるまで繰り返す．ただし，職業ユーザを 100 人以上収集できなかったとき，その職業を職業カテゴリから除外する．これは，職業のメタデータが付与されたツイートを十分に確保することができず，その職業を典型的使用場面として持つ単語を取得することが困難と考えたためである．除外した具体的な職業については 4.2.1 項で報告する．

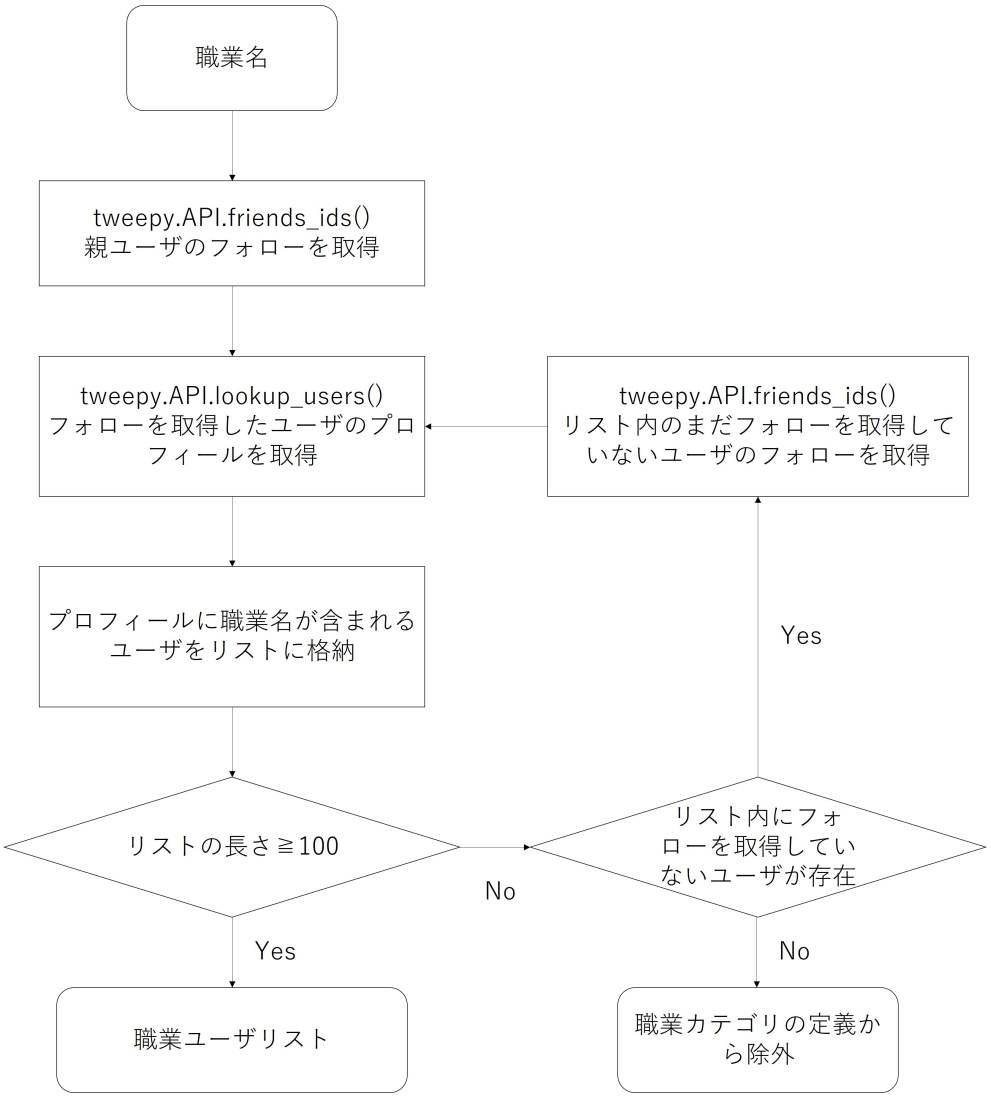

図 3.2: 職業ユーザリストを取得するフローチャート

例として職業カテゴリ「医師」のユーザを取得する過程を図 3.3 に示す．まず，親ユーザがフォローしかつ職業が医師であるユーザ A，ユーザ C を取得する．さ

らに，ユーザ C がフォローしかつ職業が医師であるユーザ D，ユーザ F を取得する．このような操作を 100 名の職業ユーザが得られるまで繰り返す．

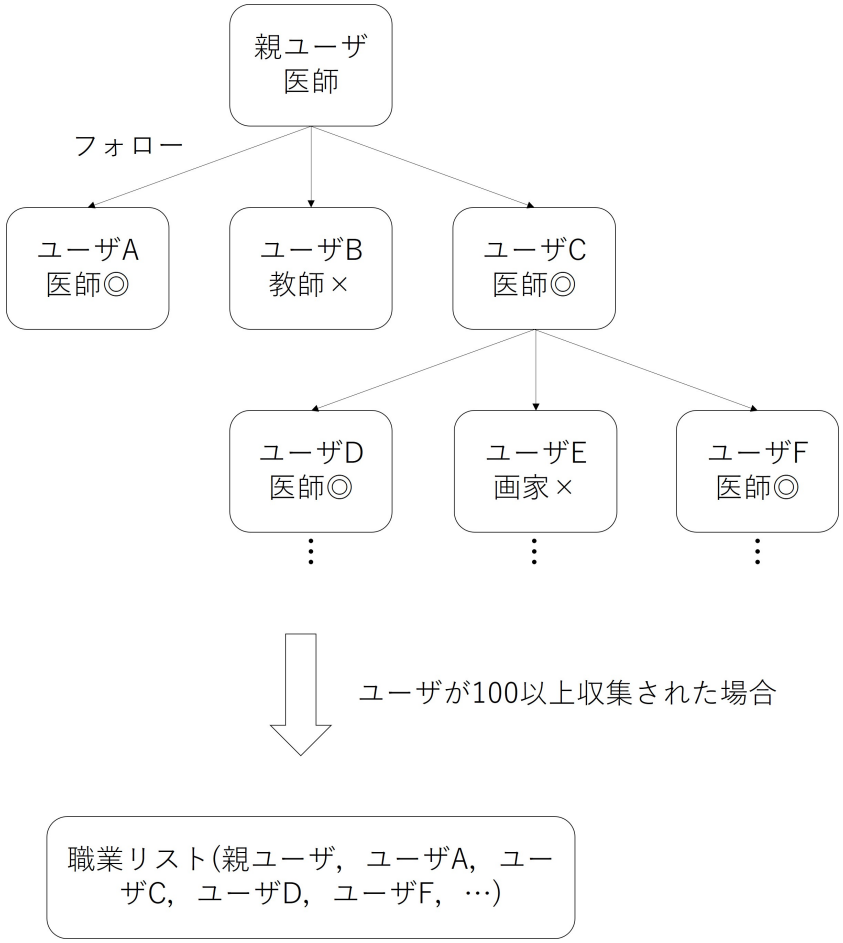

図 3.3: 職業カテゴリ「医師」のユーザの取得

最後に，職業のカテゴリがメタデータとして付与されたツイートを収集する．職業カテゴリのそれぞれについて，その職業ユーザリストに登録されているユーザが投稿した過去の全てのツイートを tweepy.API.user timeline() を用いて取得する．全ての職業カテゴリについて 100 名のユーザのツイートを取得するが，一人のユーザから得られるツイート数は異なるため，職業カテゴリが付与されたツイートの数もカテゴリによって異なる．

3.2.4 時間のツイートの収集

本研究における時間のカテゴリは【朝】【昼】【夕方】【夜】【深夜】である．Twitter では，全てのツイートには年・日付・時刻といった投稿時間のメタデータが付与されている．そのため，ツイートを上記の 5 つのカテゴリに分類するのは容易であ

る．本研究では 3.2.2 項で収集した場所のツイートのデータを流用し，そのツイートに投稿時間によって【朝】【昼】【夕方】【夜】【深夜】のいずれかのカテゴリをメタデータとして付与する．職業のツイートにも投稿時間のメタデータは付与されているが，1 つのカテゴリにつき 100 名程度のユーザが投稿したツイートで構成されているため，投稿内容に偏りがある恐れがある．そのため，職業カテゴリが付与されたツイートのデータは時間のツイートデータとして流用しない．

3.3 候補単語の抽出

収集したツイートから典型的使用場面付き辞書に登録する候補単語を抽出する．Tweepy によって取得されるツイートのデータは JSON 形式であり，ツイート本文以外にもメタデータなどの様々な情報を含むため，取得したデータからツイート本文のみを取り出す．また，ツイートのテキスト本体にも本研究では不要な情報が含まれているため，これらを除去する処理を行う．以上の前処理の詳細は 3.3.1 項で述べる．次に，ツイート本文を形態素解析によって単語に分割し，分割された単語の中から辞書に登録するべき候補単語を選別する．この詳細は 3.3.2 項で述べる．なお，これらの処理は時間・場所・職業の全てのカテゴリについて共通である．

3.3.1 前処理

一般的に Twitter でツイートを投稿し，ウェブブラウザ上で確認すると図 3.4 のように表示される．このツイートを Tweepy で取得すると，図 3.5 に示すような JSON フォーマットのデータが得られる．図 3.4，図 3.5 の例は本研究のために開設した Twitter のアカウントから投稿したツイートである[7]．また，ツイートのテキスト内のリプライ先 (@CTTM51k0iyouDCC) も本研究用の Twitter のアカウントである．以降の処理のため，Tweepy で取得される JSON 形式のデータから，ツイート本文と主要なメタデータのみを抽出して整形する．整形後のツイート例を表 3.4 に示す．

[7]https://twitter.com/CTTM51k0iyouDCC/status/1219708428953669633

図 3.4: ウェブブラウザ上で確認したツイートの例

Status(_api=<tweepy.api.API object at 0x000001C0EEFD7CC0>, _json={'created_at': 'Tue Jan 21 19:48:45 +0000 2020', 'id': 1219708428953669633, 'id_str': '1219708428953669633', 'full_text': '@CTTM51k0iyouDCC 昨日、ラーメンを食べに行った。https://hogehogehoge #ラーメン #外食 $hoge', 'truncated': False, 'display_text_range': [0, 68], 'metadata': {'iso_language_code': 'ja', 'result_type': 'recent'}, 'source': 'Twitter for Android', 'in_reply_to_status_id': None, 'in_reply_to_status_id_str': None, 'in_reply_to_user_id': 1047319349055242240, 'in_reply_to_user_id_str': '1047319349055242240', 'in_reply_to_screen_name': 'CTTM51k0iyouDCC', 'user': {'id': 1047319349055242240, 'id_str': '1047319349055242240', 'name': 'T', 'screen_name': 'CTTM51k0iyouDCC', 'location': '', 'description': '', 'url': None, 'entities': {'description': {'urls': []}}, 'protected': False, 'followers_count': 1, 'friends_count': 1, 'listed_count': 0, 'created_at': 'Wed Oct 03 02:56:05 +0000 2018', 'favourites_count': 0, 'utc_offset': None, 'time_zone': None, 'geo_enabled': True, 'verified': False, 'statuses_count': 23, 'lang': None, 'contributors_enabled': False, 'is_translator': False, 'is_translation_enabled': False, 'profile_background_color': 'F5F8FA', 'profile_background_image_url': None, 'profile_background_image_url_https': None, 'profile_background_tile': False, 'profile_image_url': 'http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', 'profile_image_url_https': 'https://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', 'profile_link_color': '1DA1F2', 'profile_sidebar_border_color': 'C0DEED', 'profile_sidebar_fill_color': 'DDEEF6', 'profile_text_color': '333333', 'profile_use_background_image': True, 'has_extended_profile': False, 'default_profile': True, 'default_profile_image': True, 'following': False, 'follow_request_sent': False, 'notifications': False, 'translator_type': 'none'}, 'geo': None, 'coordinates': None, 'place': {'id': '9271351bfef45457', 'url': 'https://api.twitter.com/1.1/geo/id/9271351bfef45457.json', 'place_type': 'city', 'name': 'Nomi-shi', 'full_name': 'Nomi-shi, Ishikawa', 'country_code': 'JP', 'country': 'Japan', 'contained_within': [], 'bounding_box': {'type': 'Polygon', 'coordinates': [[[136.426059, 36.388609], [136.631912, 36.388609], [136.631912, 36.472131], [136.426059, 36.472131]]]}, 'attributes': {}}, 'contributors': None, 'is_quote_status': False, 'retweet_count': 0, 'favorite_count': 0, 'favorited': False, 'retweeted': False, 'lang': 'ja'}, created_at=datetime.datetime(2020, 1, 21, 19, 48, 45), id=1219708428953669633, id_str='1219708428953669633', full_text='@CTTM51k0iyouDCC 昨日、ラーメンを食べに行った。https://hogehogehoge #ラーメン #外食 $hoge', truncated=False, display_text_range=[0, 68], metadata={'iso_language_code': 'ja', 'result_type': 'recent'}, source='Twitter for Android', source_url='http://twitter.com/download/android', in_reply_to_status_id=None, in_reply_to_status_id_str=None, in_reply_to_user_id=1047319349055242240, in_reply_to_user_id_str='1047319349055242240', in_reply_to_screen_name='CTTM51k0iyouDCC', author=User(_api=<tweepy.api.API object at 0x000001C0EEFD7CC0>, _json={'id': 1047319349055242240, 'id_str': '1047319349055242240', 'name': 'T', 'screen_name': 'CTTM51k0iyouDCC', 'location': '', 'description': '', 'url': None, 'entities': {'description': {'urls': []}}, 'protected': False, 'followers_count': 1, 'friends_count': 1, 'listed_count': 0, 'created_at': 'Wed Oct 03 02:56:05 +0000 2018', 'favourites_count': 0, 'utc_offset': None, 'time_zone': None, 'geo_enabled': True, 'verified': False, 'statuses_count': 23, 'lang': None, 'contributors_enabled': False, 'is_translator': False, 'is_translation_enabled': False, 'profile_background_color': 'F5F8FA', 'profile_background_image_url': None, 'profile_background_image_url_https': None, 'profile_background_tile': False, 'profile_image_url': 'http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', 'profile_image_url_https': 'https://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', 'profile_link_color': '1DA1F2', 'profile_sidebar_border_color': 'C0DEED', 'profile_sidebar_fill_color': 'DDEEF6', 'profile_text_color': '333333', 'profile_use_background_image': True, 'has_extended_profile': False, 'default_profile': True, 'default_profile_image': True, 'following': False, 'follow_request_sent': False, 'notifications': False, 'translator_type': 'none'}, id=1047319349055242240, id_str='1047319349055242240', name='T', screen_name='CTTM51k0iyouDCC', location='', description='', url=None, entities={'description': {'urls': []}}, protected=False, followers_count=1, friends_count=1, listed_count=0, created_at=datetime.datetime(2018, 10, 3, 2, 56, 5), favourites_count=0, utc_offset=None, time_zone=None, geo_enabled=True, verified=False, statuses_count=23, lang=None, contributors_enabled=False, is_translator=False, is_translation_enabled=False, profile_background_color='F5F8FA', profile_background_image_url=None, profile_background_image_url_https=None, profile_background_tile=False, profile_image_url='http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', profile_image_url_https='https://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', profile_link_color='1DA1F2', profile_sidebar_border_color='C0DEED', profile_sidebar_fill_color='DDEEF6', profile_text_color='333333', profile_use_background_image=True, has_extended_profile=False, default_profile=True, default_profile_image=True, following=False, follow_request_sent=False, notifications=False, translator_type='none'), user=User(_api=<tweepy.api.API object at 0x000001C0EEFD7CC0>, _json={'id': 1047319349055242240, 'id_str': '1047319349055242240', 'name': 'T', 'screen_name': 'CTTM51k0iyouDCC', 'location': '', 'description': '', 'url': None, 'entities': {'description': {'urls': []}}, 'protected': False, 'followers_count': 1, 'friends_count': 1, 'listed_count': 0, 'created_at': 'Wed Oct 03 02:56:05 +0000 2018', 'favourites_count': 0, 'utc_offset': None, 'time_zone': None, 'geo_enabled': True, 'verified': False, 'statuses_count': 23, 'lang': None, 'contributors_enabled': False, 'is_translator': False, 'is_translation_enabled': False, 'profile_background_color': 'F5F8FA', 'profile_background_image_url': None, 'profile_background_image_url_https': None, 'profile_background_tile': False, 'profile_image_url': 'http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', 'profile_image_url_https': 'https://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', 'profile_link_color': '1DA1F2', 'profile_sidebar_border_color': 'C0DEED', 'profile_sidebar_fill_color': 'DDEEF6', 'profile_text_color': '333333', 'profile_use_background_image': True, 'has_extended_profile': False, 'default_profile': True, 'default_profile_image': True, 'following': False, 'follow_request_sent': False, 'notifications': False, 'translator_type': 'none'}, id=1047319349055242240, id_str='1047319349055242240', name='T', screen_name='CTTM51k0iyouDCC', location='', description='', url=None, entities={'description': {'urls': []}}, protected=False, followers_count=1, friends_count=1, listed_count=0, created_at=datetime.datetime(2018, 10, 3, 2, 56, 5), favourites_count=0, utc_offset=None, time_zone=None, geo_enabled=True, verified=False, statuses_count=23, lang=None, contributors_enabled=False, is_translator=False, is_translation_enabled=False, profile_background_color='F5F8FA', profile_background_image_url=None, profile_background_image_url_https=None, profile_background_tile=False, profile_image_url='http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', profile_image_url_https='https://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png', profile_link_color='1DA1F2', profile_sidebar_border_color='C0DEED', profile_sidebar_fill_color='DDEEF6', profile_text_color='333333', profile_use_background_image=True, has_extended_profile=False, default_profile=True, default_profile_image=True, following=False, follow_request_sent=False, notifications=False, translator_type='none'), geo=None, coordinates=None, place=Place(_api=<tweepy.api.API object at 0x000001C0EEFD7CC0>, id='9271351bfef45457', url='https://api.twitter.com/1.1/geo/id/9271351bfef45457.json', place_type='city', name='Nomi-shi', full_name='Nomi-shi, Ishikawa', country_code='JP', country='Japan', contained_within=[], bounding_box=BoundingBox(_api=<tweepy.api.API object at 0x000001C0EEFD7CC0>, type='Polygon', coordinates=[[[136.426059, 36.388609], [136.631912, 36.388609], [136.631912, 36.472131], [136.426059, 36.472131]]]), attributes={}), contributors=None, is_quote_status=False, retweet_count=0, favorite_count=0, favorited=False, retweeted=False, lang='ja')

図 3.5: Tweepy で取得されるツイートの例

表 3.4: 整形したツイートの例

	ツイートのテキスト（メタデータ）
ツイート 1	@abcde 僕は朝ご飯をいっぱい食べた。 https://abcde #朝ご飯 #朝食 $abcde （ツイート ID：12345，投稿者 ID：13579，...）
ツイート 2	@qwert 今日はよく頑張った。 https://qwert #疲れた #頑張った $qwert （ツイート ID：54321，投稿者 ID：24680，...）
ツイート 3	@abcde 僕は朝ご飯をいっぱい食べた。 https://abcde #朝ご飯 #朝食 $abcde （ツイート ID：12345，投稿者 ID：13579，...）
ツイート 4	@abcde 僕は朝ご飯をいっぱい食べた。 https://abcde #朝ご飯 #朝食 $abcde （ツイート ID：98765，投稿者 ID：13579，...）
ツイート 5	@qwert 今日はよく頑張った。 https://qwert #疲れた #頑張った $qwert （ツイート ID：12321，投稿者 ID：36912，...）

収集したツイートの中には全く同じものが含まれることがある．表 3.4 の例では，ツイート 1 とツイート 3 はツイート ID が同じであり，全く同じツイートである．これは，tweepy.API.search() の制限のため，10 日間毎にプログラムを実行してツイートを収集したため，収集開始日や終了日あたりに投稿されたツイートは重複して取得されることがあるためである．このようなツイートは統合し，1 つのツイートとする．また，ツイート ID は異なっていても，ツイートのテキストが同一である場合もある．表 3.4 におけるツイート 1 とツイート 4 は，ツイート ID は異なるが，テキストと投稿者 ID が同じである．これは，bot などの何度も同じツイートをするユーザに多く見られる．このようなツイートも 1 つのツイートに統合する．

ツイートのテキストには，ハッシュタグなど，Twitter 固有の記号でマークアップされる自然言語文以外の要素も含まれる．次のステップではツイートの形態素解析を行うが，その前にこれらの要素をあらかじめ除去する．除去する対象となる要素は以下の 3 つである．

URL https//…で始まる文字列を除去する．

ユーザ名 (@) Twitter では，ユーザ名は@realDonaldTrump のように@で始まる文字列で表わされ，ツイートに対する返信先を明示するためなどに使われる．これを除去する．

**ティッカーシンボル$** ティッカーシンボルとは，株式市場における上場企業を表わす記号であり，$GOOG のように$で始まる文字列で表わされる．これを除去する．

テキスト内にはハッシュタグ（#）も存在するがこの時点では除去しない．理由は後述する．表 3.4 のツイートに対して前処理をした後のツイートを表 3.5 に示す．

表 3.5: 前処理済みのツイートの例

	ツイートのテキスト（メタデータ）
ツイート 1	僕は朝ご飯をいっぱい食べた。 #朝ご飯 #朝食 （ツイート ID：12345，投稿者 ID：13579，...）
ツイート 2	今日はよく頑張った。 #疲れた #頑張った （ツイート ID：54321，投稿者 ID：24680，...）
ツイート 5	今日はよく頑張った。 #疲れた #頑張った （ツイート ID：12321，投稿者 ID：36912，...）

3.3.2 形態素解析・候補単語の選別

前処理済みのツイートに対して形態素解析を行い，単語に分割する．形態素解析にはオープンソースの形態素解析エンジンである MeCab[8]を用いる．次に，辞書に登録するべき候補単語として，文法的機能を表わす機能語ではなく，何らかの意味を持つ自立語を選別する．具体的には，品詞が名詞・動詞・形容詞・副詞である単語を選別する．ただし，MeCab では，数字のみや記号のみの単語が名詞として分類されることがあるが，これらは候補単語として不適切なので，選別しない．また，候補単語を抽出する際には，活用形ではなく原形を取得する．単語の活用形を原形に直す処理も MeCab を用いる．ただし，MeCab では，英語表記の単語などは原形が出力されない．このときは出現形が原形に相当するので，出現形を候補単語として抽出する．また，MeCab では，辞書に登録されていない未知語に対して何らかの品詞が推定される仕様になっているため，未知語や新語に関しても，その品詞が名詞・動詞・形容詞・副詞のいずれかに推定されれば，辞書の候補単語に含まれる．

本研究では，ハッシュタグも典型的使用場面が付与された辞書の登録候補単語とする．Twitter を対象とした自然言語処理の応用システムでは，典型的使用場面の付与されたハッシュタグも有用と考えられるためである．ただし，ハッシュタグは形態素解析せずにそのまま候補単語として抽出する．すなわち，#とそれに続く文字列をそのまま候補単語とする．

[8]https://github.com/taku910/mecab

辞書の候補単語を選別後，それぞれの候補単語の統計情報を得る．具体的には，獲得したツイート集合において，以下の 3 つの統計情報をカウントする．

単語出現頻度 (単語数) ツイート集合における単語の出現頻度 (のべ数) をカウントする．

ツイート数 ツイート集合において，その単語を含むツイートの数をカウントする．

ユーザ数 ツイート集合において，その単語を含むツイートを発信したユーザの数をカウントする．

表 3.6 は，抽出された候補単語ならびにそれらの単語数，ツイート数，ユーザ数の例である．

表 3.6: 抽出された候補単語の例

候補単語	単語数	ツイート数	ユーザ数
難波千日前	110	110	98
長堀橋	140	137	97
高井田	118	115	95
ヤンマースタジアム	108	108	91
#ユニバ	139	139	96
SUNHALL	100	99	90
北堀江	123	119	91
ステーションシティシネマ	114	114	87
南方	108	104	88
住之江公園	194	186	88

3.4 典型的使用場面の特定

前節で取得した候補単語に対し，それが典型的に使用される時間，場所，職業を特定する．基本的に，候補単語とカテゴリ (時間，場所，職業のいずれか) の相関関係を定量化し，それが十分に高いとき，典型的使用場面付き辞書に登録する．単語とカテゴリの相関関係を測る手法として，自己相互情報量 (Pointwise MutualInforation: PMI) 基づく手法と，Kleinberg のバースト検出アルゴリズムに基づく手法の 2 つを提案する．前者を 3.4.1 項で，後者を 3.4.2 項で説明する．

3.4.1 自己相互情報量による特定

自己相互情報量 (PMI) は，2 つの変数の共起の強さを測る統計的指標である．ここでは，自己相互情報量を用いて単語とカテゴリの相関の強さ (共起の強さ) を測り，単語の典型的使用場面を特定する．以下，時間・場所または職業のカテゴリを c，候補単語を w とする．式 (3.1) は自己相互情報量によって計算されるスコアである．

$$PMI(w,c) = -\ln \frac{P(w|c)}{P(w)} \tag{3.1}$$

$$P(w|c) = \frac{\text{カテゴリ } c \text{ のツイートにおける単語 } w \text{ の出現頻度}}{\text{カテゴリ } c \text{ のツイートにおける全単語の出現頻度}} \tag{3.2}$$

$$P(w) = \frac{\text{単語 } w \text{ の出現頻度}}{\text{全単語の出現頻度}} \tag{3.3}$$

$P(w|c)$ は，カテゴリ c であるという条件下での候補単語 w の出現確率である．一方，$P(w)$ は，データセット全体における候補単語 w の出現確率である．式 (3.1) は $P(w|c)$ と $P(w)$ の比である．c と w の強さは $P(w|c)$ だけでも測れるが，どのカテゴリにもよく出現する単語 ($P(w)$ が大きい単語) はカテゴリとの相関の強さに関係なく $P(w|c)$ が大きくなる傾向がある．自己相互情報量では，$P(w|c)$ を $P(w)$ で割ることにより，このような単語のスコアが過度に高くなることを抑制している．

3.4.2 Kleinberg のバースト検知による特定

Kleinberg のバースト検知アルゴリズム [9] を基に，候補単語の典型的使用場面を特定する．Kleinberg のバースト検知は，時系列データ付きのキーワード集合において特定のキーワードの使用が急激に増加することを検知するアルゴリズムである．本研究では，カテゴリの列を仮想的な時系列とみなし，特定の時間帯 (本研究の場合はカテゴリ) に単語の使用頻度が急激に増加することを検出し，検出されたカテゴリをその単語の典型的使用場面とする．時間，場所または職業のカテゴリを c，候補単語を w とするとき，w と c の関連度の強さのスコアを $\sigma^t(0, r_c^t, d_c^t)$ とする．その定義を式 (3.4) に示す．

$$\sigma^t(0, r_c^t, d_c^t) = -\ln \left[\binom{d_c^t}{r_c^t} p_0^{r_c^t} (1-p_0)^{d_c^t - r_c^t} \right] \tag{3.4}$$

$$\text{ただし，} p_0 = \frac{R^t}{D^t},\ R^t = \sum_{c \in C} r_c^t,\ D^t = \sum_{c \in C} d_c^t$$

r_c^t はカテゴリが c で候補単語 w を含むツイートの数，d_c^t はカテゴリが c であるツイートの数，C はカテゴリの集合である．p_0 はデータセット全体における候補

単語 w の平均出現確率である．カテゴリ c における候補単語 w の出現確率が p_0 よりも大きいほど，$\sigma^t(0, r_c^t, d_c^t)$ は大きい値をとる．

図 3.6 は，Kleinberg の指標の性質を示すグラフである．このグラフの縦軸の値を X とするとき，$-\ln X$ が $\sigma^t(0, r_c^t, d_c^t)$ のスコアに相当するが，対数の性質から，X の値が小さいほど (グラフ上の値が小さいほど) スコアが大きくなることに注意していただきたい．このグラフは，$\frac{r_c^t}{d_c^t}$ が p_0 から離れれば離れるほどスコアが大きくなることを意味する．p_0 は，カテゴリを区別せず，データセット全体においてある単語がツイートに出現する確率である．ある特定のカテゴリ c のツイート集合における単語の出現確率がデータ全体の平均の出現確率よりも大きく異なるとき，スコアが高く算出されるようになっている．

次に，以下の 2 つの条件を満たす単語を典型的使用場面付き辞書に登録する単語として選択する．

$$\sigma^t(0, r_c^t, d_c^t) > K \tag{3.5}$$

$$\frac{r_c^t}{d_c^t} > \frac{R^t}{D^t} (= p_0) \tag{3.6}$$

式 (3.5) は，$\sigma^t(0, r_c^t, d_c^t)$ が閾値 K よりも大きいという条件である．閾値 K は，どのカテゴリについても，辞書に登録される単語が 50 個以上となるように設定する．また，時間，場所，職業のそれぞれについて，閾値 K を別々に設定する．一方，式 (3.6) は，$\frac{r_c^t}{d_c^t}$ が p_0 よりも大きいという条件である．図 3.6 に示すように，$\sigma^t(0, r_c^t, d_c^t)$ は，ある単語がカテゴリ c のツイートにあまり出現しないとき ($\frac{r_c^t}{d_c^t}$ が p_0 よりも小さいとき) にも高く見積られる．本研究ではカテゴリと関連性の強い単語を検出したいので，式 (3.6) の条件を設けた．

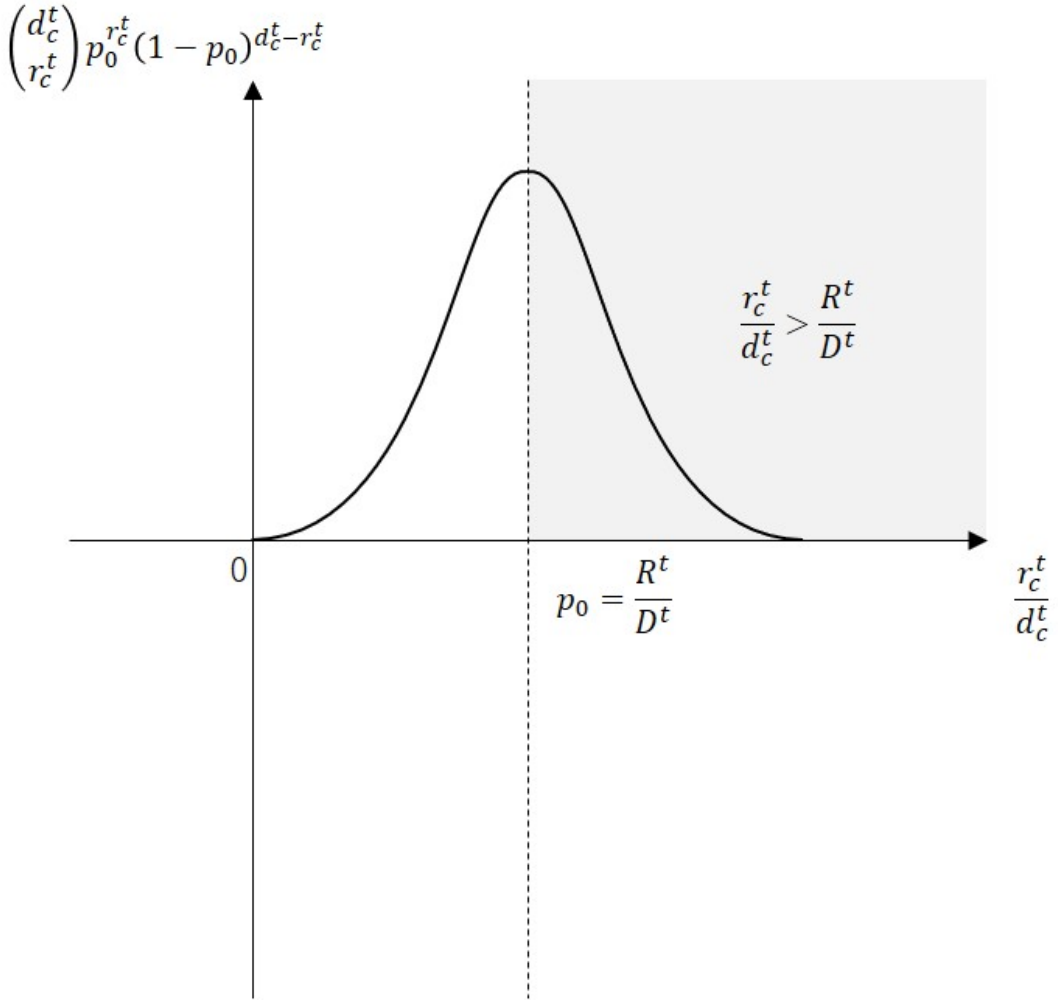

図 3.6: Kleinberg のバースト検知の特性

以上の手法では，ツイートを単位として，単語がある特定のカテゴリによく出現するかを特定している．つまり，ある単語があるカテゴリのツイートによく出現するとき，そのカテゴリを典型的使用場面のカテゴリとして特定する．ところが，同一ユーザが繰り返し同じ単語を使う場合には，ある単語を含むツイートの数が多くても，その単語がカテゴリと関連が深いとは言えないことがある．例えば，あるユーザが深夜に「今日のゲームはこれで終わり」と毎日ツイートしているとき,「ゲーム」という単語は【深夜】というカテゴリによく出現するが,「ゲーム」の時間の典型的使用場面は深夜とは言い難い．

このような問題を解決するために，別の手法として，ユーザを単位として Kleinberg のバースト検出アルゴリズムを適用する方法を提案する．ここでは，あるカテゴリにおいて，ある単語を含むツイートを発信しているユーザ数が多いとき，そのカテゴリを単語の典型的使用場面として特定する．つまり，あるカテゴリにおいて，その単語を含むツイート数ではなく，その単語を使うユーザ数が多いこと

を検出する．これにより，ある場面 (カテゴリ) で一人のユーザが同じ単語を繰り返し使う場合，その単語のユーザ数は 1 なので，その単語とカテゴリの関連度のスコアが高く見積もられる可能性が低い．具体的には，式 (3.7)，式 (3.8)，式 (3.9) に示す条件で，典型的使用場面付きの辞書に登録する単語を選択する．

$$\sigma^u(0, r_c^u, d_c^u) = -\ln\left[\binom{d_c^u}{r_c^u} p_0^{r_c^u}(1-p_0)^{d_c^u - r_c^u}\right] \tag{3.7}$$

$$\text{ただし，} p_0 = \frac{R^u}{D^u},\ \ R^u = \sum_{c\in C} r_c^u,\ \ D^u = \sum_{c\in C} d_c^u$$

$$\sigma^u(0, r_c^u, d_c^u) > K \tag{3.8}$$

$$\frac{r_c^u}{d_c^u} > \frac{R^u}{D^u} (= p_0) \tag{3.9}$$

r_c^u はカテゴリが c で候補単語 w を使用したユーザの数，d_c^u はカテゴリが c であるツイートを投稿したユーザの数，C はカテゴリの集合である．p_0 はデータセット全体における候補単語 w の平均出現確率である．カテゴリ c における候補単語 w の出現確率が p_0 よりも大きいほど，$\sigma^u(0, r_c^u, d_c^u)$ は大きい値をとる．

式 (3.8) と式 (3.9) は，それぞれ式 (3.5) と式 (3.6) と同じ意味を持つ．式 (3.8) は，$\sigma^u(0, r_c^u, d_c^u)$ が閾値 K よりも大きいという条件である．閾値 K は，どのカテゴリについても，辞書に登録される単語が 50 個以上となるように設定する．また，時間，場所，職業のそれぞれについて，閾値 K を別々に設定する．一方，式 (3.9) は，$\frac{r_c^u}{d_c^u}$ が p_0 よりも大きいという条件である．$\sigma^u(0, r_c^u, d_c^u)$ は，ある単語がカテゴリ c のツイートにあまり出現しないとき ($\frac{r_c^u}{d_c^u}$ が p_0 よりも小さいとき) にも高く見積られる．本研究ではカテゴリと関連性の強い単語を検出したいので，式 (3.9) の条件を設けた．

3.5 辞書のフォーマット

各カテゴリについて，それを典型的使用場面とする単語を選別した後，それらをまとめて最終的な辞書を構築する．本研究で想定する典型的使用場面付き辞書のフォーマットを表 3.7 に示す．「スコア」は 3.4.1 項，3.4.2 項で紹介した PMI，$\sigma^t(0, r_c^t, d_c^t)$，$\sigma^u(0, r_c^u, d_c^u)$ のいずれかの値を意味している．「–」は典型的使用場面のカテゴリが特定できなかった場合を表わす．また，「外来」，「寒い」，「Python」のように，二つまたは三つのタイプで典型的使用場面のカテゴリが特定された単語もある．

表 3.7: 典型的使用場面付き辞書のフォーマット

単語	時間		場所		職業	
	カテゴリ	スコア	カテゴリ	スコア	カテゴリ	スコア
雪	-	-	北海道	80.33	-	-
#おやすみ	夜	72.98	-	-	-	-
病院	-	-	-	-	看護師	88.79
外来	朝	33.76	-	-	医師	92.68
寒い	朝	43.35	青森県	76.99	-	-
Python	-	-	東京都	62.45	プログラマ	89.11
⋮	⋮	⋮	⋮	⋮	⋮	⋮

第4章　評価

本章は，3 章で述べた提案手法の評価実験を行う．4.1 節では職業ユーザの自動取得手法を評価する．4.2 節では，提案手法により構築された辞書について報告する．4.3 節では，構築した辞書を人手によって評価する．

4.1　職業ユーザ収集の評価

3.2.3 項で述べたように，本研究では，職業がメタデータとして付与されたツイートを収集するために，与えられた職業名を職として持つ Twitter ユーザ (職業ユーザ) を半自動的に所得する．ここでは，その手法によってどれだけ正確に職業ユーザを取得できるかを評価する．比較対象とするベースライン手法は，Twitter API を用いてプロフィールに職業名を含むユーザを職業ユーザとして取得する手法とする．44 個の職業カテゴリのうち 6 つを選択し，その職業カテゴリについて取得した職業ユーザを 20 名，合計 120 名のユーザを評価データとする．評価に用いた 6 つの職業カテゴリは，4.3.4 項で後述する実験において，ツイート数の分布に対して Kleinberg のバースト検知手法を適用する手法 (3.4.2 項) によって特定された職業の典型的使用場面推定の正解率が高い上位 3 件「プログラマ」「漁師」「理学療法士」および下位 3 件「通訳案内士」「消防士」「主婦」とした．評価データの個々のユーザについて，その職業のユーザとして適切かどうかを人手で判定し，適切なユーザの割合を正解率として算出する．判定は 1 名の作業者で行う．実験結果を表 4.1 に示す．

表 4.1: 職業ユーザ収集の評価

	正解率
提案手法	0.71
ベースライン	0.62

提案手法の方がベースラインよりも正解率が高い．ベースライン手法では，たとえプロフィールに職業名があっても，bot や宣伝を目的としたユーザが含まれることが多かった．これに対し，提案手法では，人手によって適切と判定した親ユーザがフォローしているユーザを辿って職業ユーザを取得するため，bot や宣伝ではない適切なユーザが得られることが多かった．

4.2 辞書の構築

4.2.1 職業ユーザ取得の結果

3.2 節で述べたように，職業の典型的使用場面を持つ辞書を構築する際には，職業カテゴリ毎に，それを職業とする Twitter ユーザのリストを取得する．提案手法によって取得された職業ユーザの数を表 4.2 に示す．

最初は表 3.3 に示す 50 個の職業カテゴリを設定していたが，これらのうち,「マネージャー」,「司書」,「スポーツトレーナー」,「フリーター」,「植木職人」,「マッサージ師」については，職業ユーザを 100 名以上得ることができなかった．したがって，これら 6 つのカテゴリを除き，表 4.2 に示す 44 の職業カテゴリを今後の実験で用いる職業カテゴリのセットとする．提案手法では，職業ユーザが 100 名得られた時点で処理を終了するが,「保育士」,「薬剤師」,「声優」は 300 名以上のユーザが収集された．提案手法では，ある職業ユーザがフォローするユーザを全て取得するため，一度のステップでプロフィールに職業カテゴリ名が含まれるユーザが多く発見できた場合，結果として 100 名以上の職業ユーザが収集されるためである．

表 4.2: 職業カテゴリ毎に得られた職業ユーザの数

職業カテゴリ	ユーザ数	職業カテゴリ	ユーザ数
看護師	182	画家	156
保育士	312	料理人	130
医師	119	消防士	100
パティシエ	217	映画監督	154
理学療法士	271	助産師	120
薬剤師	464	通訳案内士	105
美容師	109	モデル	103
建築家	106	学芸員	101
トリマー	210	俳優	101
教師	155	客室乗務員	116
漫画家	174	税理士	160
作家	102	管理栄養士	152
声優	356	歯科衛生士	101
飼育員	106	バーテンダー	115
ホテルマン	100	自衛官	104
歌手	265	気象予報士	157
プログラマ	102	漁師	105
アナウンサー	104	ブロガー	131
弁護士	100	芸人	268
カメラマン	100	書道家	109
議員	134	学生	125
システムエンジニア	100	主婦	101

4.2.2 ツイート収集の結果

3.2 節の手法を用いて，2019 年 1 月から 12 月にかけて，時間，場所，職業のメタデータが付与されたツイートを収集した．時間，場所，職業のメタデータが付与されたツイートのカテゴリ毎の数を表 4.3，表 4.4，表 4.5 にそれぞれ示す．場所のメタデータが付与されたツイートは 2019 年 1 月から 7 月にかけて収集し，それを用いて典型的使用場面付き辞書を構築した．さらに，継続して 2019 年 12 月までツイートを収集し，それを時間のメタデータが付与されたツイートとして利用し，時間の典型的使用場面が付与された辞書を構築した．したがって，場所のメタデータが付与されたツイートと時間のメタデータが付与されたツイートは，収集方法は同じだが，ツイートの総数は異なる．

表 4.3: 時間のメタデータ付きツイートの数

時間カテゴリ	ツイート数
深夜	1,796,716
朝	4,180,071
昼	5,202,181
夕方	3,434,670
夜	6,706,358

表 4.4: 場所のメタデータ付きツイートの数

場所カテゴリ	ツイート数	場所カテゴリ	ツイート数
北海道	434,792	滋賀県	73,822
青森県	75,449	京都府	169,950
岩手県	84,671	大阪府	694,531
宮城県	147,877	兵庫県	270,768
秋田県	52,127	奈良県	75,168
山形県	59,388	和歌山県	48,041
福島県	132,561	鳥取県	32,314
茨城県	137,750	島根県	32,705
栃木県	152,077	岡山県	122,296
群馬県	136,480	広島県	143,584
埼玉県	363,673	山口県	95,775
千葉県	404,243	徳島県	40,170
東京都	1,906,558	香川県	79,292
神奈川県	529,862	愛媛県	84,107
新潟県	139,596	高知県	41,845
富山県	60,384	福岡県	243,849
石川県	85,795	佐賀県	48,194
福井県	43,032	長崎県	61,297
山梨県	50,483	熊本県	55,514
長野県	132,435	大分県	65,312
岐阜県	99,591	宮崎県	83,300
静岡県	259,343	鹿児島県	58,538
愛知県	600,134	沖縄県	126,080
三重県	91,123		

表 4.5: 職業のメタデータ付きツイートの数

職業カテゴリ	ツイート数	職業カテゴリ	ツイート数
看護師	240,853	画家	251,569
保育士	293,679	料理人	172,188
医師	178,210	消防士	35,744
パティシエ	132,202	映画監督	232,988
理学療法士	394,555	助産師	121,105
薬剤師	624,438	通訳案内士	132,675
美容師	241,229	モデル	191,443
建築家	193,807	学芸員	146,035
トリマー	193,211	俳優	203,692
教師	163,818	客室乗務員	78,378
漫画家	335,092	税理士	175,579
作家	178,151	管理栄養士	202,547
声優	654,514	歯科衛生士	121,928
飼育員	110,284	バーテンダー	141,014
ホテルマン	107,468	自衛官	118,678
歌手	455,434	気象予報士	229,683
プログラマ	157,805	漁師	138,315
アナウンサー	180,741	ブロガー	290,816
弁護士	145,052	芸人	624,380
カメラマン	171,583	書道家	140,773
議員	241,565	学生	113,279
システムエンジニア	165,335	主婦	78,236

次に，3.3.1 項の手法を用いてツイートに対して前処理を行った．時間，場所，職業について，前処理後のカテゴリ別のツイート数を表 4.6，表 4.7，表 4.8 にそれぞれ示す．表 4.3 と表 4.6，表 4.4 と表 4.7，表 4.5 と表 4.8 を比較すると，どのカテゴリもツイート数が減少している．これは，前処理によって不適切なツイートが除去されたためである．

表 4.6: 時間のメタデータ付きツイートの数 (前処理後)

時間カテゴリ	ツイート数
深夜	1,746,746
朝	3,819,606
昼	4,994,293
夕方	3,290,114
夜	6,452,543

表 4.7: 場所のメタデータ付きツイートの数 (前処理後)

場所カテゴリ	ツイート数	場所カテゴリ	ツイート数
北海道	415,956	滋賀県	70,302
青森県	71,839	京都府	162,025
岩手県	81,632	大阪府	654,397
宮城県	142,329	兵庫県	254,637
秋田県	50,265	奈良県	70,887
山形県	57,139	和歌山県	45,922
福島県	124,829	鳥取県	30,200
茨城県	131,677	島根県	31,194
栃木県	143,489	岡山県	117,565
群馬県	128,935	広島県	136,321
埼玉県	337,203	山口県	91,880
千葉県	380,212	徳島県	38,337
東京都	1,787,930	香川県	75,842
神奈川県	494,838	愛媛県	80,884
新潟県	132,890	高知県	40,717
富山県	58,036	福岡県	234,080
石川県	80,099	佐賀県	45,729
福井県	40,800	長崎県	59,449
山梨県	48,301	熊本県	53,923
長野県	127,310	大分県	62,436
岐阜県	92,928	宮崎県	79,854
静岡県	247,296	鹿児島県	56,937
愛知県	560,917	沖縄県	121,163
三重県	87,742		

表 4.8: 職業のメタデータ付きツイートの数 (前処理後)

職業カテゴリ	ツイート数	職業カテゴリ	ツイート数
看護師	162,340	画家	238,279
保育士	280,478	料理人	143,808
医師	147,583	消防士	23,543
パティシエ	87,886	映画監督	220,335
理学療法士	377,774	助産師	103,122
薬剤師	580,608	通訳案内士	114,604
美容師	227,107	モデル	180,068
建築家	183,618	学芸員	139,598
トリマー	187,128	俳優	198,520
教師	155,863	客室乗務員	69,554
漫画家	324,287	税理士	170,983
作家	165,749	管理栄養士	167,866
声優	597,945	歯科衛生士	87,179
飼育員	102,823	バーテンダー	109,733
ホテルマン	93,697	自衛官	104,858
歌手	400,580	気象予報士	223,702
プログラマ	128,387	漁師	123,677
アナウンサー	173,533	ブロガー	278,780
弁護士	125,746	芸人	594,646
カメラマン	166,510	書道家	127,107
議員	230,997	学生	77,009
システムエンジニア	139,492	主婦	71,784

最後に，表 4.9 に前処理済みのツイートの総数，カテゴリ毎のツイート数のうち最大および最小のツイート数，カテゴリ当たりの平均ツイート数を示す．時間のメタデータが付与されたツイートの総数が約 2 千万件と最も多い．場所，職業のメタデータが付与されたツイートの総数はほぼ同じで，850 万件程度であった．また，カテゴリ毎のツイート数を見ると，最大値と最小値の差が大きいことから，収集されたツイート数はカテゴリによってばらつきがある．

表 4.9: 収集したツイートの概要

	時間	場所	職業
総数	20,303,302	8,439,273	8,608,886
最大	6,452,543【夜】	1,787,930【東京都】	597,945【声優】
最小	1,746,746【深夜】	30,200【鳥取県】	23,543【消防士】
平均	4,060,660	179,559	195,657

4.2.3 候補単語抽出の結果

表 4.6，表 4.7，表 4.8 に示したツイート集合に対して，3.2.2 項の手法を用いて典型的使用場面の辞書に登録する候補単語を抽出した．時間，場所，職業のそれぞれについて，抽出した候補単語数を表 4.10，表 4.11，表 4.12 にそれぞれ示す．

表 4.10: 時間の辞書の候補単語数

時間カテゴリ	候補単語数
深夜	522,280
朝	822,116
昼	1,088,793
夕方	822,602
夜	1,231,232

表 4.11: 場所の辞書の候補単語数

場所カテゴリ	候補単語数	場所カテゴリ	候補単語数
北海道	160,109	滋賀県	59,736
青森県	54,718	京都府	108,281
岩手県	63,098	大阪府	248,715
宮城県	88,939	兵庫県	134,164
秋田県	47,208	奈良県	60,307
山形県	51,680	和歌山県	45,817
福島県	79,740	鳥取県	32,167
茨城県	82,730	島根県	34,661
栃木県	85,643	岡山県	77,009
群馬県	81,671	広島県	84,869
埼玉県	154,141	山口県	62,850
千葉県	166,137	徳島県	41,115
東京都	548,235	香川県	59,567
神奈川県	207,117	愛媛県	61,216
新潟県	84,193	高知県	42,657
富山県	54,631	福岡県	125,003
石川県	67,186	佐賀県	44,637
福井県	40,951	長崎県	53,150
山梨県	47,725	熊本県	49,185
長野県	84,685	大分県	52,288
岐阜県	69,203	宮崎県	62,840
静岡県	123,839	鹿児島県	52,792
愛知県	208,050	沖縄県	84,273
三重県	65,944		

表 4.12: 職業の辞書の候補単語数

職業カテゴリ	候補単語数	職業カテゴリ	候補単語数
看護師	70,539	画家	115,041
保育士	85,707	料理人	83,608
医師	98,145	消防士	23,146
パティシエ	48,011	映画監督	122,432
理学療法士	123,496	助産師	62,644
薬剤師	175,438	通訳案内士	94,279
美容師	99,624	モデル	98,961
建築家	108,374	学芸員	96,050
トリマー	72,224	俳優	78,953
教師	70,192	客室乗務員	51,109
漫画家	132,496	税理士	75,251
作家	104,634	管理栄養士	81,313
声優	170,698	歯科衛生士	59,097
飼育員	53,639	バーテンダー	67,032
ホテルマン	70,150	自衛官	75,404
歌手	173,924	気象予報士	99,281
プログラマ	80,588	漁師	63,468
アナウンサー	107,097	ブロガー	113,718
弁護士	76,231	芸人	183,710
カメラマン	71,792	書道家	85,506
議員	107,080	学生	53,660
システムエンジニア	92,844	主婦	58,586

表 4.13 に，時間，場所，職業のそれぞれについて，候補単語の総数，カテゴリ毎の候補単語数のうち最大および最小の候補単語数，カテゴリ当たりの平均候補単語数を示す．ツイート数については，場所・職業のツイートに比べて時間のツイートが多かったが，候補単語数は時間，場所，職業とで大きな差はなかった．また，カテゴリ毎の候補単語数については，最大と最小との差が大きいことから，ツイート数と同様にカテゴリによってばらつきが見られる．また，表 4.9 と表 4.13 を比較すると，最大のツイート数を収集した職業カテゴリは【声優】，最大の候補単語数が得られた職業カテゴリは【芸人】と異なっている．これ以外は，最大・最小のカテゴリは，ツイート数と候補単語数とで一致している．

表 4.13: 候補単語の概要

	時間	場所	職業
総数	4,487,023	4,394,872	4,035,172
最大	1,231,232【夜】	548,235【東京都】	183,710【芸人】
最小	522,280【深夜】	32,167【鳥取県】	23,146【消防士】
平均	89,740.60	93,507.91	91,708.46

4.2.4 典型的使用場面付き辞書構築の予備実験

前節までの処理で得られた候補単語について，3.4.1 項で述べた PMI スコアによって，場所カテゴリに特有の単語を選別する予備実験を行った．それぞれの場所カテゴリについて，PMI のスコアの上位 20 件の単語を選択した．これらの単語について，その場所カテゴリが典型的使用場面として適切かどうかを人手で評価した．判定は 1 名の作業者が実施した．実際に得られた単語の例として,【石川県】の PMI の上位 20 件の単語と人手による判定結果を表 4.14 に示す.「正解判定」の列は，正解と判定した場合は 1，不正解と判定した場合は 0 を表わす．正解と判定した単語は 5 つと少なかった．また，全ての単語の PMI のスコアの値は同じであり，多くの単語に対してスコアが同点となっていた．

表 4.14: 石川県の PMI の特定上位 20 項目

単語	PMI	正解判定
#濡れ豆	4.62	1
#camperlife	4.62	0
Kalpa	4.62	0
Relinquish	4.62	0
#便通	4.62	0
レモンチューハイ	4.62	0
#能登ワイン	4.62	1
#人間科学部教授	4.62	0
ニューギニアダト	4.62	0
#地獄の面談	4.62	0
#mr2spyder	4.62	0
PPFM	4.62	1
#画質悪し	4.62	0
#チャンカレの L カツ	4.62	1
人臣	4.62	0
#パレットランチ	4.62	0
コジラ	4.62	0
#今年は波ある年	4.62	0
#今日はブラック派	4.62	0
Hodatsushimizu	4.62	1

次に，それぞれの場所カテゴリについて，PMI スコアにより選別した上位 20 件の単語の正解率を求める．正解率の定義を式 (4.1) に示す．同様に，全カテゴリ (47 都道府県) についての正解率も求める．場所カテゴリ別ならびに全カテゴリの正解単語数と正解率を表 4.15 に示す．

$$\text{正解率} = \frac{\text{正解単語数}}{\text{判定した単語の数}} \tag{4.1}$$

表 4.15: PMI によって選別した場所カテゴリに関連の深い単語の評価

場所カテゴリ	正解単語数	正解率	場所カテゴリ	正解単語数	正解率
北海道	3	0.15	滋賀県	5	0.25
青森県	10	0.50	京都府	4	0.20
岩手県	3	0.15	大阪府	6	0.30
宮城県	8	0.40	兵庫県	3	0.15
秋田県	6	0.30	奈良県	4	0.20
山形県	4	0.20	和歌山県	4	0.20
福島県	2	0.10	鳥取県	5	0.25
茨城県	5	0.25	島根県	4	0.20
栃木県	7	0.35	岡山県	5	0.25
群馬県	3	0.15	広島県	6	0.30
埼玉県	11	0.55	山口県	2	0.10
千葉県	10	0.50	徳島県	5	0.25
東京都	10	0.50	香川県	3	0.15
神奈川県	4	0.20	愛媛県	5	0.25
新潟県	5	0.25	高知県	4	0.20
富山県	5	0.25	福岡県	6	0.30
石川県	5	0.25	佐賀県	2	0.10
福井県	6	0.30	長崎県	4	0.20
山梨県	5	0.25	熊本県	1	0.05
長野県	5	0.25	大分県	13	0.65
岐阜県	7	0.35	宮崎県	3	0.15
静岡県	1	0.05	鹿児島県	7	0.35
愛知県	4	0.20	沖縄県	9	0.45
三重県	5	0.25			

	正解数	正解率
全場所カテゴリ	244	0.26

表 4.15 に示すように，カテゴリ毎の正解率は全体的に低く，全カテゴリの正解率も 0.26 と非常に低い．低頻度の単語があるカテゴリのみに含まれる場合，その単語とカテゴリに対する PMI は大きくなる性質がある．そのため，たまたまあるカテゴリのみに出現した低頻度の単語が誤って多く抽出されたと考えられる．以上の実験結果を踏まえ，PMI によってカテゴリと関連の深い単語を選択して典型的使用場面付き辞書を構築することは適切ではないと判断した．

4.2.5 典型的使用場面付き辞書構築の結果

3.4.2 項で述べた Kleinberg のバースト検知アルゴリズムに基づく手法によってカテゴリと関連の深い単語を特定し，典型的使用場面付き辞書を構築した．本研究では，典型的使用場面付き辞書を構築する以下の 2 つの手法を提案した．

- ツイート数の分布に対して Kleinberg のバースト検知手法を適用する手法

  個々の単語について，その単語が使用されたツイート数のカテゴリ毎の分布を求め，これに対して特定のカテゴリでよく使われる (バーストする) 単語を Kleinberg の手法により特定する手法である．具体的には，式 (3.4)，(3.5)，(3.6) を用いて，各カテゴリに特有の単語を選別し，典型的使用場面付き辞書を構築する．以下，この手法を「Kleinberg-tweet 手法」と呼ぶ．

- ユーザ数の分布に対して Kleinberg のバースト検知手法を適用する手法

  個々の単語について，その単語が使用したユーザ数のカテゴリ毎の分布を求め，これに対して特定のカテゴリでよく使われる (バーストする) 単語を Kleinberg の手法により特定する手法である．具体的には，式 (3.7)，(3.8)，(3.9) を用いて，各カテゴリに特有の単語を選別し，典型的使用場面付き辞書を構築する．以下，この手法を「Kleinberg-user 手法」と呼ぶ．

Kleinberg-tweet 手法によって構築された辞書の単語ののべ数，異り数，カテゴリ数，1 カテゴリ当たりの平均単語数，閾値 K を表 4.16 に示す．本研究の提案手法では，1 つの単語に対して複数のカテゴリが割り当てられることがある．単語ののべ数とは，ここでは各カテゴリに割り当てられた単語の総数であり，単語の異り数とは，複数のカテゴリに登録された単語を 1 つと数えたときの単語の数，すなわち単語の種類の数である．閾値 K は，Kleinberg の手法のスコアの閾値であり，この値よりスコアが大きい単語を辞書に登録する．既に述べたように，全てのカテゴリで少なくも 50 個の単語を辞書に登録するように閾値 K を決定した．

表 4.16: Kleinberg-tweet 手法で構築された辞書の概要

	時間	場所	職業
単語のべ数	1,717	17,132	7,509
単語異り数	1,715	16,554	7,440
カテゴリ数	5	47	44
平均単語数	343.40	364.51	170.66
閾値 K	24.00	47.25	101.66

表 4.16 に示すように，時間，場所，職業のいずれにおいても，単語のべ数よりも単語異り数が小さい．これは，複数のカテゴリが割り当てられた単語が存在することを意味する．また，時間と場所の辞書は職業の辞書よりも閾値 K が小さい

が，カテゴリ当たりの平均単語数が多い．一般に，閾値 K が大きくなると辞書に登録される単語は少なくなるが，今回の実験では，時間と場所の辞書については，閾値以上のスコアを持つ単語が多く獲得されたことがわかった．

同様に，Kleinberg-user 手法によって構築された辞書の概要を表 4.17 に示す．

表 4.17: Kleinberg-user 手法で構築された辞書の概要

	時間	場所	職業
単語のべ数	1,152	6,793	199,475
単語異り数	1,149	6,512	103,481
カテゴリ数	5	47	44
平均単語数	230.40	144.53	4,533.52
閾値 K	13.73	52.22	6.86

Kleinberg-user 手法においても，表 4.17 に示すように，時間，場所，職業のいずれにおいても，単語のべ数よりも単語異り数が小さい．やはり複数のカテゴリが割り当てられた単語が存在することが確認された．また，閾値 K とカテゴリ当たりの平均単語数の関係も，K が小さいときに平均単語数が小さくなる傾向は見られない．特に職業の辞書では，閾値 K が一番小さいにも関わらず，平均単語数が 4,533,52 と大きい．職業のメタデータが付与されたツイートは，表 4.2 に示したように 100〜464 名のユーザから収集しており，時間や場所の辞書よりもユーザ数が少ない．Kleinberg-user 手法は，ユーザ数の分布に対して Kleinberg のバースト検知アルゴリズムを適用しているため，ユーザ数が少ないことが影響していると考えられる．

4.3 辞書の評価

本節では，Kleinberg-tweet 手法，Kleinberg-user 手法によって構築された典型的使用場面付き辞書を評価する．

4.3.1 実験の手順

構築された辞書に登録された単語から，カテゴリ毎に，Kleinberg の手法によって算出されたスコアの上位 20 件の単語を選択し，評価データとする．評価データの個々の単語について，そのカテゴリが単語の典型的使用場面として適切かどうかを人手で判定する．判定は 2 名の作業者で行う．

辞書の評価基準は，カテゴリ毎もしくは辞書全体での正解率とする．正解率の定義を式 (4.2) に示す．

$$\text{正解率} = \frac{\text{作業者 1 の正解単語数} + \text{作業者 2 の正解単語数}}{\text{評価対象単語数} \times 2} \tag{4.2}$$

この正解率は，作業者 2 名の判定結果をまとめて算出することに注意していただきたい．カテゴリの正解率を算出するときは，あるカテゴリについて上位 20 件の単語を 2 名の作業者が判定するため，評価対象とする単語ののべ数は $20 \times 2 = 40$ であり，正解率はその 40 個の中で正解と判定された単語の割合となる．また，2 名の作業者の判定の一致度を評価する κ 係数を算出し，正解判定の揺れを評価する．

判定作業の例として，場所のカテゴリのひとつである【石川県】について，Kleinberg-tweet 手法で構築された辞書のうちスコアの上位 20 件の単語，その単語を含むツイート数，Kleinberg の手法によるスコア，2 者の判定結果を表 4.18 に示す．「作業者 1」「作業者 2」の列は，それぞれの作業者が正解と判定したときは 1，不正解と判定したときは 0 を記す．同様に，Kleinberg-user 手法で【石川県】のカテゴリが付与された単語の評価結果を表 4.19 に示す．

表 4.18: Kleinberg-tweet 手法により【石川県】のカテゴリが付与された単語の評価

単語	ツイート数	スコア	作業者 1	作業者 2
#輪島	87	321.23	1	1
藤江	89	313.34	1	1
バイパスレジャーランド	84	309.24	1	1
#ペンションベッセル	81	300.17	1	1
#民泊	81	273.51	0	0
内灘	71	263.43	1	1
和倉温泉	74	260.12	1	1
光浦	75	256.05	1	1
穴水	69	251.30	1	1
#アルプラザ金沢	65	241.38	1	1
#ティアラ	66	241.21	0	0
金澤	89	239.87	1	1
金沢	71	234.65	1	1
#E7 系	66	231.68	0	1
#selectshop	85	228.80	0	0
#E7 系運用	60	223.00	0	1
加賀温泉	62	210.29	1	1
#KANAZAWA	55	204.62	1	1
アヘシ	57	201.20	0	0
かがやく	81	194.39	0	0

表 4.19: Kleinberg-user 手法により【石川県】のカテゴリが付与された単語の評価

単語	ユーザ数	スコア	作業者 1	作業者 2
七尾	149	565.50	1	1
Ishikawa	129	492.25	1	1
近江	153	409.50	0	1
小松	122	400.00	1	1
能登	109	382.25	1	1
能登	120	367.50	1	1
千里浜	102	364.50	1	1
フォーラス	88	320.50	1	1
#石川県	85	306.00	1	1
片町	83	278.75	1	1
白山	100	270.50	1	1
羽咋	70	265.25	1	1
香林坊	66	249.13	1	1
松任	65	249.00	1	1
輪島	67	246.88	1	1
津幡	60	229.00	1	1
和倉温泉	62	228.63	1	1
藤江	57	209.63	0	1
バイパスレジャーランド	54	207.88	1	1
石川	120	205.75	1	1

4.3.2 時間の典型的使用場面付き辞書の評価

Kleinberg-tweet 手法による時間の典型的使用場面付き辞書の評価の結果を表 4.20 に示す．Kleinberg-user 手法による時間の典型的使用場面付き辞書の評価の結果を表 4.21 に示す．

表 4.20: Kleinberg-tweet 手法による時間の典型的使用場面付き辞書の評価

カテゴリ	正解数	正解率
深夜	20	0.50
朝	13	0.33
昼	21	0.53
夕方	20	0.50
夜	14	0.35

表 4.21: Kleinberg-user 手法による時間の典型的使用場面付き辞書の評価

カテゴリ	正解数	正解率
深夜	21	0.53
朝	35	0.88
昼	18	0.45
夕方	11	0.28
夜	33	0.83

Kleinberg-tweet 手法では，表 4.20 に示すように，カテゴリ毎の正解率は 0.33 から 0.53 と低かった．一方，Kleinberg-user 手法では，表 4.21 に示すように,【夕方】の正解率は 0.28 と低いものの,【朝】【夜】の正解率は 80%を越えた．全体的に見て，Kleinberg-user 手法の方が Kleinberg-tweet 手法よりも正解率が高かった．

Kleinberg-tweet 手法でスコアの高かった単語について考察する．正解と判定された単語は，その時間帯に放送されるテレビ番組，ラジオ番組の名称やそれに関連する単語が多かった．例えば，カテゴリ【深夜】では「#CDTV」，カテゴリ【昼】では「#ほんサタ」が得られた．また，正解の単語の多くはハッシュタグであった．その理由として，Twitter では，現在放送されているテレビ番組，ラジオ番組に対するコメントをハッシュタグを付けて投稿するユーザが多いと考えられる．また，カテゴリ【朝】では「#朝のご挨拶」，カテゴリ【昼】では「#コースランチ」といったように，テレビ番組，ラジオ番組に関連しない単語もわずかに見られた．正解率の最も低いカテゴリは【朝】であったが，例えば「#東武練馬床屋」が誤って獲得された．この単語は，1 名のユーザが同じ時間帯に数多く投稿しているためにスコアが高く検出された．このように，一人のユーザが同じ時間帯に同じ単語を繰り返し投稿したときに，不正解の単語が獲得されることが多かった．そのため，1 名のユーザの偏った投稿の影響を小さくするための対策のひとつが，Kleinberg-user 手法のようにユーザ数の分布によってスコアを計算する方法である．

Kleinberg-user 手法でスコアの高かった単語について考察する．正解と判定した単語は，Kleinberg-tweet 手法と同様に，テレビ番組，ラジオ番組に関連する単語が多かった．例えば，カテゴリ【昼】では「#のど自慢」，カテゴリ【夜】では「#SOL」が得られた．また，カテゴリ【朝】では「#朝風呂」，カテゴリ【昼】では「#ランチタイム」といったように，番組以外の適切な単語も，Kleinberg-tweet 手法と比べて数多く獲得できた．カテゴリ【朝】については,「オハヨウ」,「ぐっども」のような朝の挨拶を短縮・簡略化した表現が見られた．正解率の最も低いカテゴリは【夕方】であったが，不正解の単語として「#じゅわチキ」などが見られた．この単語は，食品の企業が自社の商品の販売を促進することを目的に,「#じゅわチキ」というハッシュタグをつけて投稿することをユーザに促していた．また，本研究では，どのカテゴリについても最低 50 個の単語が得られるように Kleinberg

のスコアの閾値を設定したが，時間の辞書ではこれが低く設定された．結果として，Kleinberg のスコアの値が低い単語も獲得されており，候補単語の中にカテゴリに特徴的な単語自体が少なかったと考えられる．

4.3.3 場所の典型的使用場面付き辞書の評価

Kleinberg-tweet 手法による場所の典型的使用場面付き辞書の評価の結果を表 4.22 に示す．Kleinberg-user 手法による場所の典型的使用場面付き辞書の評価の結果を表 4.23 に示す．

表 4.22: Kleinberg-tweet 手法による場所の典型的使用場面付き辞書の評価

カテゴリ	正解数	正解率	カテゴリ	正解数	正解率
北海道	37	0.93	滋賀県	31	0.78
青森県	27	0.68	京都府	32	0.80
岩手県	36	0.90	大阪府	31	0.78
宮城県	38	0.95	兵庫県	28	0.70
秋田県	29	0.73	奈良県	31	0.78
山形県	31	0.78	和歌山県	28	0.70
福島県	36	0.90	鳥取県	32	0.80
茨城県	33	0.83	島根県	29	0.73
栃木県	35	0.88	岡山県	30	0.75
群馬県	39	0.98	広島県	33	0.83
埼玉県	31	0.78	山口県	28	0.70
千葉県	33	0.83	徳島県	18	0.45
東京都	30	0.75	香川県	28	0.70
神奈川県	35	0.88	愛媛県	29	0.73
新潟県	37	0.93	高知県	34	0.85
富山県	26	0.65	福岡県	40	1.00
石川県	28	0.70	佐賀県	30	0.75
福井県	31	0.78	長崎県	34	0.85
山梨県	33	0.83	熊本県	36	0.90
長野県	30	0.75	大分県	36	0.90
岐阜県	38	0.95	宮崎県	15	0.38
静岡県	31	0.78	鹿児島県	29	0.73
愛知県	38	0.95	沖縄県	34	0.85
三重県	31	0.78			

表 4.23: Kleinberg-user 手法による場所の典型的使用場面付き辞書の評価

カテゴリ	正解数	正解率	カテゴリ	正解数	正解率
北海道	40	1.00	滋賀県	36	0.90
青森県	38	0.95	京都府	35	0.88
岩手県	39	0.98	大阪府	39	0.98
宮城県	33	0.83	兵庫県	36	0.90
秋田県	37	0.93	奈良県	32	0.80
山形県	39	0.98	和歌山県	39	0.98
福島県	37	0.93	鳥取県	37	0.93
茨城県	36	0.90	島根県	34	0.85
栃木県	33	0.83	岡山県	37	0.93
群馬県	38	0.95	広島県	34	0.85
埼玉県	33	0.83	山口県	37	0.93
千葉県	36	0.90	徳島県	35	0.88
東京都	33	0.83	香川県	32	0.80
神奈川県	37	0.93	愛媛県	40	1.00
新潟県	39	0.98	高知県	30	0.75
富山県	38	0.95	福岡県	38	0.95
石川県	38	0.95	佐賀県	34	0.85
福井県	39	0.98	長崎県	38	0.95
山梨県	35	0.88	熊本県	35	0.88
長野県	36	0.90	大分県	39	0.98
岐阜県	38	0.95	宮崎県	31	0.78
静岡県	37	0.93	鹿児島県	39	0.98
愛知県	36	0.90	沖縄県	37	0.93
三重県	32	0.80			

Kleinberg-tweet 手法では，表 4.22 に示すように，カテゴリ毎の正解率は 0.38 から 1.00 と正解率に幅があった．一方，Kleinberg-user 手法では，表 4.23 に示すように，低いものでも【高知県】の正解率は 0.75,【宮城県】の正解率は 0.78 であり，その他の場所の正解率は 80%を越えた．全体的に見て，Kleinberg-user 手法の方が Kleinberg-tweet 手法よりも正解率が高かった．

Kleinberg-tweet 手法でスコアの高かった単語について考察する．正解と判定した単語は，どのカテゴリにおいても，地名やその場所に存在する施設名が多かった．例えば，カテゴリ【北海道】では「小樽」，カテゴリ【宮城県】では「#仙台駅」が獲得された．また，カテゴリ【青森県】では「よごす」，カテゴリ【茨城県】では「ロボッツ」といったように，地名・施設名以外にも，方言，地元のスポーツチーム名の単語も見られた．正解率の最も低いカテゴリは【徳島県】であったが，

不正解と判定した単語に「#10 秒小説」があった．不正解の単語の多くは，時間の辞書で考察したように，1 名のユーザが同じ単語を含むツイートを数多く投稿している場合に獲得されると考えられる．

Kleinberg-user 手法でスコアの高かった単語について考察する．正解と判定した単語は，Kleinberg-tweet 手法と同じように，どのカテゴリにおいても地名や施設名が多かった．例えば，カテゴリ【福井県】では「Fukui」，カテゴリ【三重県】では「#伊勢神宮」が得られた．また，地名・施設名以外でも，カテゴリ【青森県】では「ねぶた」，カテゴリ【山口県】では「レノファ」といったように，地元の祭り，地元のスポーツチーム名の単語も見られた．正解率の最も低いカテゴリは【高知県】であったが，誤って獲得された単語に「ひろめる」,「市場」があった．これらの単語は形態素解析の誤りによって獲得された．高知県の観光名所に「ひろめ市場」があるが，これが「ひろめ」[1]と「市場」に誤分割され,「ひろめる」「市場」が獲得されたと考えられる．

4.3.4 職業の典型的使用場面付き辞書の評価

Kleinberg-tweet 手法による職業の典型的使用場面付き辞書の評価の結果を表 4.24 に示す．Kleinberg-user 手法による職業の典型的使用場面付き辞書の評価の結果を表 4.25 に示す．

[1]「ひろめる」の連用形

表 4.24: Kleinberg-tweet 手法による職業の典型的使用場面付き辞書の評価

カテゴリ	正解数	正解率	カテゴリ	正解数	正解率
看護師	16	0.40	画家	27	0.68
保育士	18	0.45	料理人	22	0.55
医師	15	0.38	消防士	7	0.18
パティシエ	27	0.68	映画監督	23	0.58
理学療法士	31	0.78	助産師	28	0.70
薬剤師	23	0.58	通訳案内士	11	0.28
美容師	31	0.78	モデル	20	0.50
建築家	19	0.48	学芸員	20	0.50
トリマー	31	0.78	俳優	15	0.38
教師	12	0.30	客室乗務員	28	0.70
漫画家	21	0.53	税理士	21	0.53
作家	15	0.38	管理栄養士	23	0.58
声優	22	0.55	歯科衛生士	15	0.38
飼育員	27	0.68	バーテンダー	21	0.53
ホテルマン	14	0.35	自衛官	29	0.73
歌手	13	0.33	気象予報士	22	0.55
プログラマ	33	0.83	漁師	33	0.83
アナウンサー	23	0.58	ブロガー	19	0.48
弁護士	22	0.55	芸人	20	0.50
カメラマン	21	0.53	書道家	17	0.43
議員	19	0.48	学生	25	0.63
システムエンジニア	13	0.33	主婦	6	0.15

表 4.25: Kleinberg-user 手法による職業の典型的使用場面付き辞書の評価

カテゴリ	正解数	正解率	カテゴリ	正解数	正解率
看護師	35	0.88	画家	37	0.93
保育士	36	0.90	料理人	39	0.98
医師	39	0.98	消防士	38	0.95
パティシエ	40	1.00	映画監督	40	1.00
理学療法士	38	0.95	助産師	39	0.98
薬剤師	39	0.98	通訳案内士	35	0.88
美容師	36	0.90	モデル	38	0.95
建築家	33	0.83	学芸員	36	0.90
トリマー	39	0.98	俳優	38	0.95
教師	39	0.98	客室乗務員	39	0.98
漫画家	38	0.95	税理士	37	0.93
作家	40	1.00	管理栄養士	37	0.93
声優	39	0.98	歯科衛生士	38	0.95
飼育員	39	0.98	バーテンダー	40	1.00
ホテルマン	40	1.00	自衛官	37	0.93
歌手	40	1.00	気象予報士	40	1.00
プログラマ	40	1.00	漁師	40	1.00
アナウンサー	38	0.95	ブロガー	39	0.98
弁護士	39	0.98	芸人	34	0.85
カメラマン	39	0.98	書道家	39	0.98
議員	39	0.98	学生	22	0.55
システムエンジニア	39	0.98	主婦	0	0.00

Kleinberg-tweet 手法では，表 4.24 に示すように，カテゴリ毎の正解率は 0.15 から 0.83 と正解率に幅があった．一方，Kleinberg-user 手法では，表 4.25 に示すように,【学生】の正解率は 0.55,【主婦】の正解率は 0.00 と低いものの，その他の職業の正解率は 80%を越えた．全体的に見て，Kleinberg-user 手法の方が Kleinberg-tweet 手法よりも正解率が高かった．

Kleinberg-tweet 手法でスコアが高かった単語について考察する．正解と判定した単語は，どのカテゴリにおいても，職場で扱う道具や，その職業の業界の専門用語が多かった．例えば，カテゴリ【プログラマ】では「#PHP」，カテゴリ【歯科衛生士】では「むし歯」が得られた．また，カテゴリ【弁護士】における「申し立てる」のように，その職業の人がよく使う動詞や，カテゴリ【料理人】における「フェリチェリーナ」のような職場名 (この例はレストランの名称) の単語も獲得された．正解率の最も低いカテゴリは【主婦】であったが，誤った単語として「#迷子犬」などが見られた．このような不正解の単語は，時間や場所の辞書と同

様に，1 名のユーザが同じ単語を含むツイートを数多く投稿している場合に獲得されると考えられる．

Kleinberg-user 手法でスコアが高かった単語について考察する．正解と判定した単語は，どのカテゴリにおいても，職場で使う道具や職業の専門用語が多かった．例えば，カテゴリ【薬剤師】では「ジェネリック」，カテゴリ【建築家】では「施工」が得られた．その他に，カテゴリ【声優】の「演じる」のようなその職業の人がよく使う動詞や，カテゴリ【気象予報士】の「hPa」のような専門用語 (単位) も見られた．正解率の最も低いカテゴリは【主婦】であったが,「政権」などの政治関連の単語が数多く見られた．このような不正解の単語が獲得された原因として，3.2.3 項の手法によって収集された職業ユーザの偏りが考えられる．提案手法では，ある職業ユーザのフォローを辿って新たな職業ユーザを獲得するが，あるユーザが別のユーザをフォローするとき，職業が同じという理由だけでなく，同じ物事に興味があるという理由でフォローしていることも考えられる．カテゴリ【主婦】では政治に関する単語が数多く獲得されたが，主婦かつ政治に興味があるユーザが【主婦】の職業ユーザとして数多く収集されたためと考えられる．

4.3.5 辞書の評価のまとめ

Kleinberg-tweet 手法ならびに Kleinberg-user 手法の全カテゴリに対する正解率を表 4.26 に示す．Kleinberg-tweet 手法よりも Kleinberg-user 手法の方が，時間，場所，職業のいずれにおいても正解率が大きい．既に考察したように，Kleinberg-tweet 手法では一人のユーザが繰り返し使う単語がカテゴリに特有の単語と誤って判定されることが多いが，Kleinberg-user 手法では単語を使用したユーザ数を考慮しているので，そのような誤りが少ないことが原因と考えられる．

表 4.26: 提案手法の全カテゴリに対する正解率

	時間	場所	職業
Kleinberg-tweet 手法	0.44	0.79	0.52
Kleinberg-user 手法	0.59	0.90	0.92

Kleinberg-tweet 手法ならびに Kleinberg-user 手法のそれぞれについて，2 名の作業者による判定の κ 係数を表 4.27 に示す．時間，場所，職業を比較すると，職業の κ 係数が他の 2 つに比べて低い．その単語が職業に関連しているかどうかの判定は，人によって揺れが大きいと思われる．また，時間については Kleinberg-user 手法の方が Kleinberg-tweet 手法よりも κ 係数が高いが，場所と職業については Kleinberg-tweet 手法の方が高くなっている．特に場所の κ 係数には大きな差がある．なぜ手法の違いによって κ 係数が大きく異なるのかを調べることは今後の課題である．

表 4.27: 2 名の作業者による判定の κ 係数

	時間	場所	職業
Kleinberg-tweet 手法	0.76	0.77	0.53
Kleinberg-user 手法	0.83	0.41	0.49

第5章　おわりに

5.1　まとめ

本論文では，マイクロブログの Twitter からツイートを収集し，それに含まれる単語について，その単語が典型的に使われる時間や場所，その単語をよく使うユーザの職業を特定し，典型的使用場面付き辞書を自動的に構築した．

まず，典型的使用場面として，時間カテゴリ 5 種，場所カテゴリ 47 種，職業カテゴリ 44 種を定義した．これらのカテゴリがメタデータとして付与されたツイートを収集した．職業については，その職業を持つユーザを半自動的に収集し，それらのユーザが投稿したツイートを取得することで，職業のメタデータ付きツイートを得た．次に，収集したツイートに対し，重複するツイートの統合，テキスト以外のタグの除去，同一ユーザの同一内容のツイートの統合などの前処理を行った．続いて，ツイートを形態素解析し，名詞，動詞，形容詞，副詞，ハッシュタグを抽出し，典型的使用場面付き辞書の候補単語とした．これらの候補単語に対して，3 つの手法で典型的使用場面，すなわちその単語と関連の深い時間，場所，職業のカテゴリを特定した．1 つ目は自己相互情報量 PMI に基づく手法である．PMI によって単語とカテゴリの関連の強さを測り，単語の典型的使用場面を特定する．2 つ目は Kleinberg-tweet 手法である．ある単語について，その単語が使われているツイート数を個々のカテゴリ毎にカウントし，ツイート数のカテゴリ分布を得た．これに対して Kleinberg のバースト検出手法を適用し，あるカテゴリが他のカテゴリと比べてどれだけ突出してその単語を含むツイート数が多いかを評価するスコアを算出し，そのスコアが閾値以上のときに，そのカテゴリを単語の典型的使用場面として特定した．閾値は，全てのカテゴリで登録される単語が 50 個以上になるように設定した．3 つ目は Kleinberg-user 手法である．ある単語について，その単語が使われているユーザ数を個々のカテゴリ毎にカウントし，ユーザ数のカテゴリ分布を得た．その後，Kleinberg-tweet 手法と同様に，ユーザ数が他のカテゴリと比べて突出して多いカテゴリを単語の典型的使用場面として特定した．

評価実験では，まず PMI によって単語の典型的使用場面を特定する手法を評価した．47 の場所のカテゴリのそれぞれについて，PMI によって算出した関連度スコアの高い上位 20 件の単語を評価データとし，場所カテゴリが単語の典型的使用場面として適切であるかを人手で評価した．評価単語の数は 47 カテゴリ× 20 単語=940 単語であったが，それらの中でカテゴリが適切と判定した単語の数 (正解単語数) は 244，正解率は 0.26 となった．この結果から，PMI によって単語の典型

的使用場面を特定する手法は有効ではないと結論付けられた．

次に，Kleinberg-tweet 手法を評価した．各カテゴリから Kleinberg-tweet 手法で計算されたスコアの上位 20 単語を作業者 2 名によって人手で評価した．正解率は，時間カテゴリは 0.44，場所カテゴリは 0.79，職業カテゴリは 0.52 であった．正解の単語は，時間カテゴリではテレビ番組やラジオ番組，場所カテゴリでは地名や施設名，職業カテゴリでは職場で扱う道具や業界の用語が多かった．不正解の単語は，どのカテゴリについても，一人のユーザが典型的使用場面に関連しない単語を含むツイートを数多く投稿した場合に抽出された．

次に，Kleinberg-user 手法を評価した．Kleinberg-tweet 手法と同様に，各カテゴリ毎に上位 20 件の単語を作業者 2 名によって人手で評価した．正解率は，時間カテゴリは 0.59，場所カテゴリは 0.90，職業カテゴリは 0.92 であった．どのカテゴリも Kleinberg-tweet 手法と比べて正解率が向上した．これは，一人のユーザだけによって繰り返し使われる単語がカテゴリに特有の単語として検出されなくなったためと考えられる．正解の単語は，Kleinberg-tweet 手法で取得された単語と同じような単語が多かった．このことから，単語の典型的使用場面を特定するためには，単語を含むツイート数のカテゴリ毎の分布よりも，その単語を使ったユーザ数のカテゴリ毎の分布を求め，他のカテゴリに比べて極端にその数が多いカテゴリを検出する方が適していることが確認された．

最終的に，Kleinberg-user 手法を用いて，時間カテゴリが付与された単語を 1,152 個，場所カテゴリが付与された単語を 6,793 個，職業カテゴリが付与された単語を 1,152 個を含む典型的使用場面付き辞書が構築された．

5.2　今後の課題

本研究の課題について述べる．初めに，今回構築した典型的使用場面付き辞書では，登録された単語の品詞に偏りがあるという問題がある．具体的には，今回の実験で得られた単語は，そのほとんどが名詞またはハッシュタグであった．提案手法では，動詞，形容詞，副詞についても単語の典型的使用場面を特定できるが，実際に得られた数は少なかった．この問題を解決するための手法として，品詞ごとに辞書を構築することが考えられる．具体的には，候補単語を予め品詞ごとに分割し，Kleinberg-user 手法によってカテゴリと単語の関連度スコアを計算し，それぞれの上位の単語を合わせて典型的使用場面付き辞書を構築する．様々な品詞の単語を含めることで，典型的使用場面付き辞書の利便性が向上すると考える．

次に，単語の典型的使用場面の特定手法を改善する必要がある．本研究では，単語の典型的使用場面の特定に PMI と Kleinberg のバースト検知アルゴリズムを用いた．しかし，PMI に基づく手法の正解率は非常に低く，ユーザ数の分布に対して Kleinberg のバースト検知アルゴリズムを適用した手法の正解率は，場所カテゴリと職業カテゴリについては高いものの，時間カテゴリは 0.59 と低く，改善の余地がある．そのため，単語とカテゴリの相関の強さを測る別の手法を検討するべ

きである．例えば，TF-IDF による手法が考えられる．TF-IDF は，テキストに含まれる単語の重要度を表す指標である．具体的には，TF としてカテゴリ c に候補単語 w がどれほどよく出現するかを，IDF で候補単語 w が他のカテゴリにどれほど含まれないかを計算し，この 2 つの指標を乗算する．また，本研究のように，単語の出現頻度ではなく，その単語を用いたユーザ数によって TF-IDF を計算することも考えられる．

人手による評価実験についても課題が残されている．本研究では，典型的使用場面の単語が正解であるか不正解であるかを作業者 2 名によって判定した，しかし，評価者の人数が少なく，評価結果の信頼性に疑問が残る．そのため，人手による判定の被験者の数を増やすことで，より正確に構築した辞書の品質を評価することが必要である．また，今回の実験では，カテゴリ毎にスコアの上位 20 単語のみを評価対象とした．上位 20 件以下の単語は評価されていないため，辞書全体の品質は確認されていない．辞書全体の品質をより正確に評価するための方法として，構築した辞書からランダムに単語をサンプリングし，それらを人手で評価することが考えられる．

最後に，本研究で構築した典型的使用場面付き辞書の実用的な評価が今後の課題として挙げられる．すなわち，構築した辞書が自然言語処理システムでどの程度有用であるかを評価する．一例として，テキストの場面判定が挙げられる．まず，書かれた時間がわからないテキストに対して，時間の典型的使用場面付き辞書を用いてその時間を推定し，その正解率を測る．簡単な手法としては，テキストに含まれる個々の単語に対し，辞書を参照してそれに対応する時間カテゴリを集計し，頻度が一番高い時間カテゴリを推定結果として出力する．場所や職業についても，同様に辞書を用いて推定し，その正解率を測る．このとき，本研究で採用したメタデータ付きのツイートを収集する手法で，辞書構築に用いたツイートとは別のツイートを新たに取得してテストデータとする．このような実験は比較的容易に実施できると考えられる．

今後，以上で述べた課題を克服し，より良質な典型的使用場面付き辞書の構築に取り組みたいと考えている．

参考文献

[1] 荒牧英治，増川佐知子，森田瑞樹．Twitter Catches the Flu:事実性判定を用いたインフルエンザ流行予測． 情報処理学会研究報告，2009．

[2] 服部峻．Web 知識を用いた時空間依存な対話システムの試作．電子情報通信学会技術研究報告，AI, 人工知能と知識処理，110(105)，pp.13-18，2010．

[3] 池田定博，大橋正和，金田重郎．流行ことば・流行コンセプト予測手法．同志社政策科学研究，3(1)，pp.35-56，2002．

[4] 池田和史，服部元，松本一則，小野智弘，東野輝夫．マーケット分析のための Twitter 投稿者プロフィール推定手法．情報処理学会論文誌，2(1)，pp.82-93，2012．

[5] 石井健一.「強いつながり」と「弱いつながり」の SNS―個人情報の開示と対人関係の比較―．情報通信学会誌，29(3)，pp.25-36，2011．

[6] 自由国民社．1998 年版現代用語の基礎知識．自由国民社，1998．

[7] 自由国民社．1999 年版現代用語の基礎知識．自由国民社，1999．

[8] 川野覚，溝渕昭二．Q & A サイトを対象にした地域別土産物情報収集ツール．情報科学技術フォーラム講演論文集，14(2)，pp.221-222，2015．

[9] Jon Kleinberg．Bursty and Hierarchical Structure in Streams．Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining，pp.91-101，2002．

[10] 松田耕史，佐々木彬，岡崎直観，乾健太郎．場所参照表現タグ付きコーパスの構築と評価．情報処理学会研究報告 自然言語処理（NL），2015-NL-220(12)，pp.1-10，2015．

[11] 奥村学．ソーシャルメディアを対象としたテキストマイニング．電子情報通信学会 基礎・境界ソサイエティ Fundamentals Review 6(4)，pp.285-293，2013．

[12] 奥谷貴史，山名早人．メンション情報を利用した Twitter ユーザープロフィール推定．日本データベース学会和文論文誌，13-j(1)，pp.1-6，2014．

[13] Daniel Preotiuc-Pietro，Vasileios Lampos，Nikolaos Aletras．An analysis of the user occupational class through Twitter content．Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing，Volume 1，pp.1754-1764，2015．

[14] James Pustejovsky，José Castaño，Robert Ingria，Roser Saurí，Robert Gaizauskas，Andrea Setzer，and Graham Katz．TimeML: Robust Specification of Event and Temporal Expressions in Text．New Directions in Question Answering，Papers from 2003 AAAI Spring Symposium，Stanford University，Stanford，CA，USA，2003．

[15] James Pustejovsky，Patrick Hanks，Roser Sauri，Andrew See，Robert Gaizauskas，Andrea Setzer，Dragomir Radev，Beth Sundheim，David Day，Lisa Ferro and Marcia Lazo．The TimeBank corpus．2003

[16] 榊剛史，松尾 豊．ソーシャルセンサとしての Twitter：ソーシャルセンサは物理センサを凌駕するか?．人工知能学会誌，27(1)，pp.67-74，2012．

[17] 高橋和子，高村大也，奥村学．機械学習とルールベースによる職業コーディング．情報処理学会研究報告自然言語処理（NL），159，pp.53-60，2004．

[18] 保田 祥，小西 光，浅原 正幸，今田 水穂，前川 喜久雄.『現代日本語書き言葉均衡コーパス』に対する時間情報表現・事象表現間の時間的順序関係アノテーション．言語処理学会，20(2)，pp.201-221．2013．

付 録A　都道府県の地名コード

都道府県のメタデータが付与されたツイートを収集するために用いた地名コードを表 A.1 に示す.

表 A.1: 都道府県の地名コード

都道府県	地名コード	都道府県	地名コード
北海道	0b89db31d164a17d	滋賀県	287821abb712dd3b
青森県	1de05e90db6fde15	京都府	d4255b2b43cbf2cc
岩手県	516fad81ed9abcc2	大阪府	84316acd652607fa
宮城県	1cff59592a4767e9	兵庫県	46cd5ede80186a9c
秋田県	975c45ff265eb77c	奈良県	6836153322ac8f20
山形県	4a9a5111024ffa58	和歌山県	631253d45931eb36
福島県	5e921369a11e38d5	鳥取県	4bb7c88397417f82
茨城県	cb7c6e9092251aa1	島根県	fd9c584b35c83605
栃木県	9452db4fb01f0432	岡山県	b9f3fc68dd8f717b
群馬県	00a8aa111d38316c	広島県	39834ee320359393
埼玉県	6eb3dcfadbbe0c68	山口県	ab43b3b8a7593bb0
千葉県	7562529145a9ed1f	徳島県	4efce255445dc26a
東京都	a56612250c754f23	香川県	9411fa3e127a9e37
神奈川県	5f3279ed753778b7	愛媛県	df55059f8045566b
新潟県	5af9c3e8dadd043d	高知県	00af8e922b6236ea
富山県	c005c6ef5d97c9da	福岡県	684cc1cfd89cacef
石川県	1d059cea3e433d3d	佐賀県	6b52871b45b5b261
福井県	d3bdee61e7cfba0c	長崎県	59856611bf9bfb97
山梨県	a20dcb31ad69d661	熊本県	a160a0ba64b9b2b8
長野県	f28ae4f6babdb2b5	大分県	84166be9996a2df5
岐阜県	a3e6429d33900d31	宮崎県	82564af6cbb58e75
静岡県	d80142cda25d6767	鹿児島県	fc60aa4eb7499eb1
愛知県	c68b1ffd6bd34468	沖縄県	052e049119fd8da1
三重県	f9170e3707e30162		