

屋外における To-Do 解決支援システムの提案 A Proposal for To-Do Support System in Outdoors

鈴木 礼治[†] 松野 省吾[‡] 大山 実[†]
Reiji Suzumura Shogo Matsuno Minoru Ohyama

1. はじめに

近年、急速に普及しているスマートフォンやウェアラブルデバイスに搭載されている多様なセンサを用いて位置情報などのユーザのコンテキストを収集し、ユーザの場所に依りてタスクを通知するシステムの研究[1]や、スケジュール解析と組み合わせた行動支援システムの研究が活発に行われている[2][3]。しかし、スケジュールに比べ時間的制約少ない To-Do を対象とした研究は少ない。

既存のスマートフォンで To-Do を管理するアプリケーションには、To-Do 入力時にその To-Do が解決できる施設や場所を追加で指定し、そこに近づくと通知を行う機能がある。しかし、これを裏返して言えば、指定した施設や場所に近づかなければ通知が行われないということである。また、To-Do にはコンビニエンスストアなど、場所にもよるが、ユーザの近辺に複数個所に存在する施設で解決できるものもあると考えられる。しかし、既存のアプリケーションでは 1 つの施設、場所にしか対応していない。これは To-Do を解決する機会の損失につながると筆者らは考える。

本稿では、To-Do 入力時のユーザによる To-Do の解決場所の追加入力なしに、To-Do を解決できる施設を推定し、その施設にユーザが接近した場合に通知を行い、To-Do の解決を支援する To-Do 解決支援システムを提案する。提案システムは、Android スマートフォン上で動作し、通勤時や通学時といった屋外における使用を想定している。システムを実装したアプリケーションを開発し、その評価を行ったので報告する。

2. 提案システムの概要

提案システムは Android スマートフォン上での動作を想定している。システムを実装したアプリケーションを開発した。アプリケーションは図 1、図 2 に示す画面から構成されている。



図 1 メイン画面



図 2 設定画面

2.1 メイン画面

図 1 メイン画面では、主に To-Do の入力、表示、削除を行う。図 1 の①のテキストボックスに To-Do を入力し図 1 の②のボタンをタップすることで To-Do がシステムに入力される。To-Do が入力されると、システムはその To-Do がどの施設で解決できるかを推定する。これを 4 章で説明する。入力された To-Do は図 1 の③の To-Do 表示エリアにリスト形式で表示される。4 章で説明する To-Do の通知が完了すると、図 1 の④の“[リマインド済み]”が To-Do の右に追加される。その To-Do を長押しすると削除確認のダイアログが表示され、ユーザの同意により To-Do を削除する。図 1 の⑤のボタンから図 2 の設定画面に移動する。

2.1.1 To-Do の保存

提案システムは、表 1 に示す To-Do に関する情報を、To-Do 入力時に端末内のデータベースに保存する。表 1 の ID は To-Do を一意に識別する数値でデータベースによって自動的に付与される。表 1 の To-Do はユーザが入力した To-Do、解決可能施設は 4 章で述べる To-Do が解決できる と推定された施設である。

表 1 To-Do の DB への保存形式

To-Do 情報	例
ID	2
To-Do	本を借りる
解決可能施設	library

2.2 設定画面

図 2 設定画面では、提案システムの動作の基準となる設定をユーザが指定する。設定には以下の項目がある（番号は図 2 のものと対応している）。

- ① 施設の種類…提案システムで対応している 71 種類の施設の中からユーザが利用する、または通知して欲しい施設を指定する。71 種類の施設は Google Places API^aのプレイスタイル表^bの中から、割り振られる基準が把握できない施設 (establishment など) を除外した施設を選択した。
- ② 通知間隔…通知を行う間隔を秒単位で指定する。
- ③ 通知範囲…To-Do が解決できる施設がユーザから半径何メートル以内にある場合通知するかを指定する。
- ④ 取得範囲…施設情報を取得するユーザを中心とした範囲をメートルで指定する。

3. API を利用した施設情報の取得

ユーザによる To-Do の解決場所の追加入力なしに通知を行うためには、To-Do の解決場所となるユーザの周囲にある施設の名称、位置、種類等の情報を取得する必要がある。

^a <https://developers.google.com/places/>

^b https://developers.google.com/places/web-service/supported_types#table1

[†] 東京電機大学
[‡] 電気通信大学

提案システムでは、Google Places API を利用し施設情報を取得する。

3.1 施設情報の取得の概要

Google Places API のプレイス検索^cを利用し、ユーザの周囲にある施設の情報を取得する。ユーザを中心とした図 2 の④取得範囲内にある図 2 の①施設の種類で指定した施設の情報を取得する。例えば、施設の種類が“convenience_store”と“restaurant”で、取得範囲が 500 メートルの場合、ユーザを中心に半径 500 メートル以内にあるコンビニとレストランの情報を取得する。

3.2 取得する施設情報

施設毎に表 2 に示す情報を Google Places API から取得しデータベースに保存する。プレイス ID^dは API によって割り振られる施設を一意的に識別する ID である。表 2 の種類はプレイスタイプ表 1 の中から API によって割り振られる施設の種類を表す値で、To-Do が解決できると推定された施設の種類と比較し To-Do の通知を行うために使用する。種類は複数設定されている場合があり、提案システムが対応していない、即ち 71 種類以外の種類が設定されている場合はその種類を削除する。表 2 の緯度と経度は施設の位置で、ユーザと施設の距離の計算に使用する。

表 2 取得する施設情報の例

施設情報	例
プレイス ID	ChIjWd8Zly98ImARLDPRNHbinIY
名称	東京電機大学 千葉ニュータウンキャンパス
種類	university
緯度	35.7941869
経度	140.1095989

3.3 施設情報取得の判断

提案システムでは、約 30 秒毎に周辺の施設情報の取得を実施するかどうかの判断を行う。目的として、ユーザが移動した場合への対応、ユーザが図 2 の①施設の種類を変更した場合の施設情報の追加取得などが挙げられる。判断基準として、図 2 の①施設の種類と図 2 の④取得範囲を利用する。施設情報の取得を実施する際に表 3 に示す情報を履歴としてデータベースに保存する。緯度と経度はユーザの位置、種類は施設の種類、範囲は取得範囲である。この履歴情報から、過去に施設情報を取得した範囲と施設の種類を把握する。

表 3 記録する履歴情報の例

履歴情報	例
緯度	35.796526
経度	140.109379
種類	convenience_store, restaurant
範囲 (m)	500

施設情報の取得は図 3 に示すフローチャートに沿って行われる。図 3 の s はシステムが動作を開始してから経過時間 (秒) である。まず、履歴からユーザの現在位置が過去に施設情報を取得した範囲内であるかを確認する。範囲

外ならば施設情報取得を実行し履歴を保存する。範囲内の場合、該当する履歴から施設の種類を取り出し、現在の図 2 の①施設の種類と比較する。その結果、新しく追加された施設がある場合、その施設について施設情報取得を実行し履歴を保存する。

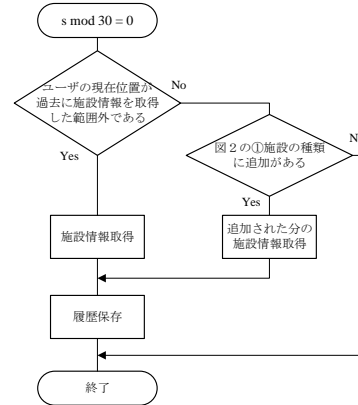


図 3 施設情報取得のフローチャート

4. To-Do 解決可能施設推定

ユーザによる To-Do の解決場所の追加入力なしに通知を行うためには、ユーザが入力した To-Do がどの施設で解決できるかを推定する必要がある。提案システムでは、ナイーブベイズ分類器を使用し推定を行う。ユーザが入力した To-Do が To-Do 解決可能施設推定によって解決できると推定された施設を解決可能施設と表記する。

4.1 教師データの収集

ナイーブベイズ分類器を用いた教師あり学習を行うには、教師データが必要になる。本来ならば、推定の対象となる To-Do を集め教師データとするべきであるが、教師データとして十分な量の 71 施設分の To-Do を集めることは困難である。そこで本研究では、提案システムが対応する 71 施設に関する Web ページの文章を教師データとした。

71 種類の施設につき、その施設に対応する Wikipedia 日本語版の Web ページ、及びその施設の日本語の名称を Google 検索で検索した結果に列挙される Web ページを収集する。Wikipedia のページが存在しない場合は Google 検索でのみ収集する。Google 検索では、検索時の環境により検索結果が変わるため、パーソナライズ検索を無効化し、google.com で検索することでローカライズ検索の影響を減少させる。また、収集した Web ページの文章に対して Java API の `java.text.Normalizer#normalize(,Normalizer.Form.NFKC)` で Unicode 正規化を行い、全角、半角の表記ゆれに対処した。Web ページ 1 件の文章を教師データの 1 文書とする。また、71 種類の施設それぞれをクラスとした。1 つの文書は 1 つのクラスにのみ属している。

4.2 学習

4.1 節で収集した教師データを用いて分類器の学習を行う。まず、形態素解析器を用いて教師データの文書を解析し文書に含まれる単語の品詞を確認する。形態素解析器として `mecab-ipadic-NEologd` [4] を辞書として使用している `Elasticsearch Analysis Kuromoji Neologd` Version 5.0.0 を利用

^c <https://developers.google.com/places/web-service/search>

^d <https://developers.google.com/places/web-service/place-id>

^e <https://github.com/codelibs/elasticsearch-analysis-kuromoji-neologd>

する。To-Do の表記法には目的語+動詞, 目的語, 動詞, 場所名の 4 つの表記法がある[5]ため, 品詞が名詞と動詞の単語を特徴語として学習に使用する。また, Web ページの文章とユーザが入力する To-Do とでは, 同じ意味でも違う単語を使用する場合があります, 教師データにユーザが入力する To-Do に含まれる特徴語が生起しない可能性がある。そこで, 日本語 WordNet^f[6]を利用して特徴語の同義語を検索し学習に使用することでその影響を減少させる。クラス毎に教師データの文書数, 単語 (特徴語とその同義語) とその生起数を学習する。

4.3 分類器

提案システムでは, ナイーブベイズ分類器の多項モデルを採用し解決可能施設の推定を行う。多項モデルの式を式 (1) に示す。

$$c^* = \arg \max_c P(c) \prod_{i=1}^N P(w_i|c) \dots\dots\dots (1)$$

To-Do に含まれる特徴語を基に, その To-Do がどのクラスに属するかを判断するための式で, クラス毎に To-Do がそのクラスに属する確率を求め, 確率が最大となるクラスを解決可能施設とする。分類器の学習に名詞と動詞を特徴語として使用するため, 推定に使用する To-Do も特徴語を抽出したものとする。式 (1) の c^* は解決可能施設, c はクラス, $P(c)$ はクラス c である確率, $P(w_i|c)$ はクラス c であるときに w_i が出現する確率である。 w は To-Do に含まれる特徴語の集合, w_i は To-Do に含まれる i 番目の特徴語, N は To-Do に含まれる特徴語数である。ここでは $P(c)$ を式 (2) と定義する。

$$P(c) = \frac{\text{クラス } c \text{ の文書数}}{\text{教師データの総文書数}} \dots\dots\dots (2)$$

また, $P(w_i|c)$ を式 (3) と定義する。

$$P(w_i|c) = \frac{\text{クラス } c \text{ における } w_i \text{ の生起数} + 1}{\text{クラス } c \text{ の全単語の生起数の合計} + \text{教師データの総単語数}} \dots\dots\dots (3)$$

式 (3) では, クラス c における特徴語 w_i の生起数が 0 になり, 確率が 0 になるゼロ頻度問題の影響を緩和するためにラプラス・スムージングを適用した。しかし, この場合 $P(w_i|c)$ が極端に小さくなり, 乗算時にアンダーフローを起こす可能性がある。そこで対数を取り加算化した式 (4) を採用する。

$$c^* = \arg \max_c \left(\log P(c) + \sum_{i=1}^N \log P(w_i|c) \right) \dots\dots\dots (4)$$

4.4 解決可能施設の推定

通常の場合では, 71 クラスそれぞれで To-Do が属する確率を算出し, 確率が最大になるクラスを解決可能施設とするが, この場合ユーザが利用しない施設が解決可能施設となる場合がある。提案システムでは, 図 2 の①施設の種類の指定されている施設に対応するクラスでのみ確率を算出し推定を行う。例として, 事前にユーザが選択した施設の種類の “atm”, “convenience_store”, “restaurant” の場合, この 3 施設に対応するクラスに To-Do が属する確率を算出し, 確率が最大となるクラスを解決可能施設とする。

5. To-Do の通知

提案システムでは, ユーザの位置を定期的に確認し, ユーザが入力した To-Do の解決可能施設がユーザの近くに存在する場合に通知を行い To-Do の解決を支援する。3 章で取得した施設情報と 4 章で推定した解決可能施設, そしてユーザの位置情報を組み合わせることでユーザによる To-Do の解決場所の追加入力なしに複数個所の施設について To-Do の通知を行うことを実現する。

5.1 ユーザの位置情報の取得

提案システムは, スマートフォンの GPS 機能を利用してユーザの位置情報を定期的に取得する。Android API の `android.location.LocationListener#onLocationChanged()` の Location オブジェクトからユーザの位置情報を緯度, 経度の形式で取得する。プロバイダは `LocationManager.GPS_PROVIDER` である。

5.2 ユーザの周囲にある施設のリストアップ

ユーザの周囲にある施設を把握するために, 図 2 の②通知間隔の間隔でユーザの位置と周囲の施設との距離を計算し図 2 の③通知範囲内の施設の情報を 3.2 節で作成したデータベースから取得する。ユーザの位置と周囲の施設との距離の計算には Android API の `android.location.Location#distanceBetween()` を利用する。データベースから取得した施設情報をユーザとの距離が近い順に並び替えリストアップする。リストアップで作成されるリストの例を表 4 に示す。

表 4 リストの例

プレイス ID	名称	種類	距離(m)
ChJJ2rLdy98ImARq8-R1uJJPFM	東京電機大学総合メディアセンター 千葉サテライトセンター図書館	library	59.84
ChJJwd8Zly98ImARLDPRNHbinY	東京電機大学 千葉ニュータウンキャンパス	university	146.10
ChJJOSoksSV8ImAR1Wfj7dP0PUQ	TDU いんざい イノベーション推進センター	university	284.44
ChJJXc1t3DF8ImARfIm8qeZXyeU	武西学園台南街区公園	park	351.48

5.3 スマートフォンの機能を用いた To-Do の通知

4 章で推定した解決可能施設と 5.2 節でリストアップした施設とを比較し, 解決可能施設と同じ種類の施設がリストに存在する場合ユーザにその To-Do と施設を通知する。通知は図 2 の②通知間隔で指定した間隔で更新される。解決可能施設と同じ施設がリストに複数存在する場合, ユーザとの距離が最も近い施設を通知する。リストに解決可能施設が存在する限り通知を継続する。通知は, 図 4 に示す Android の Notification と端末のバイブレーションによって行われる。Notification は, 1 つの To-Do につき 1 つ表示され, To-Do, 及び To-Do の解決可能施設と同じ種類のユーザから最も近い施設の名称が表示される。



図 4 Notification の表示例

^f <http://compling.hss.ntu.edu.sg/wnja/index.en.html>

5.3.1 施設の位置確認

Notification とバイブレーションによりユーザーに解決可能施設が付近にあることを通知するが、その施設の場所をユーザーが把握できなければ To-Do を解決することができない。提案システムでは、Android 端末にインストールされている Google マップを使用することでこの問題を解決する。Notification をタップすると提案システムは Google マップに対して Intent[§]を送信し、Google マップ上に対象施設を表示する。図 4 に示す上から 2 番目の Notification をタップした際に表示される Google マップを図 5 に示す。このようにユーザーは施設の場所を把握することができる。



図 5 マップ上で施設の位置確認

5.4 通知の完了

Notification をタップしたことにより該当する To-Do のユーザーに対する通知が完了したとみなす。通知が完了すると Notification とバイブレーションを停止する。各 To-Do の通知の状態を管理するために、表 1 に加えて表 5 に示す通知に関する情報を To-Do 毎にデータベースに保存する。表 5 の通知済みは値が 0 と 1 で To-Do が通知されたかを表す。通知が完了すると 1 となる。表 5 の通知中は値が 0 と 1 で Notification により To-Do が通知されているかを表す。Notification が表示されている間 1 となる。表 5 の通知中の施設のプレイス ID は通知中の施設を表し、Notification が表示されている間は、その Notification に表示されている施設のプレイス ID が値となる。

表 5 通知に関する情報の例

通知情報	例
通知済み	0
通知中	1
通知中の施設のプレイス ID	ChJI2rLdy98lmARq8-R1uJPFM

5.5 To-Do 通知の動作

To-Do の通知は図 6、図 7 に示すフローチャートに沿って実行される。図 6 の左のフローチャートがメインプロセス、メインプロセス内のサブプロセス“To-Do の確認”が図 6 の右のフローチャート、To-Do の確認内のサブプロセス“To-Do の通知”が図 7 のフローチャートである。図 6 の s はシステムが動作を開始してから経過時間 (秒) である。図 7 の PID は施設のプレイス ID である。

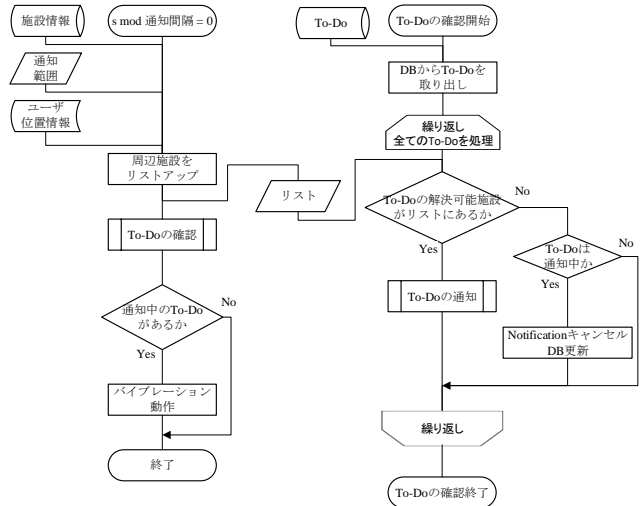


図 6 To-Do 通知のフローチャート A

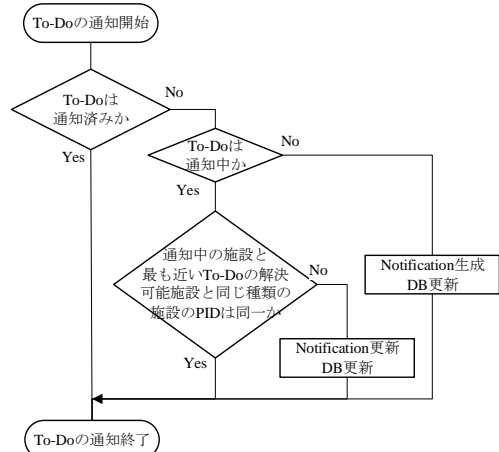


図 7 To-Do 通知のフローチャート B

6. 評価実験

本章では、提案システムの To-Do 解決可能施設推定の精度、および To-Do の通知について評価する。

6.1 To-Do 解決可能施設推定の精度

提案システムの To-Do 解決可能施設推定が、実際にユーザーが入力する To-Do をどの程度の精度で分類できるのかを実験を通して評価する。分類器の学習に使用する教師データは、2017年5月19日に収集した。

6.1.1 テストデータの収集

評価に使用するテストデータとして、25名の学生に対してアンケートを行い、166件の To-Do を収集した。アンケート回答者の負担を減らすため、提案システムが対応している 71 種類の施設のうち 34 種類の施設で解決することができる To-Do と To-Do の詳細、To-Do が解決できる施設の記述を求めた。アンケートの実際の回答を表 6 に示す。表 6 の③解決できる施設は、表 7 に示す施設-番号対応表を基に施設を番号に置き換えたものである。分類器に与える To-Do は表 6 の②To-Do である。

[§] <https://developers.google.com/maps/documentation/android-api/intents>

表 6 アンケートの回答例

①To-Doの詳細	②To-Do	③解決できる施設
ガソリンを入れる	給油	16
本を買う	本	6,12
弁当を買う	弁当	12,17,31
歯医者に行く	歯医者	13
大学へ行く	大学	34
映画を見に行く	映画	25
車を車検に出す	車検	7
飲み物を買う	水	12,17
病院に行く	病院	18
洗車をする	洗車	16

表 7 番号-施設対応表

番号	施設	番号	施設	番号	施設
1	atm	13	dentist	25	movie_theater
2	bakery	14	florist	26	park
3	bank	15	furniture_store	27	pharmacy
4	beauty_salon	16	gas_station	28	police
5	bicycle_store	17	grocery_or_supermarket	29	post_office
6	book_store	18	hospital	30	real_estate_agency
7	car_dealer	19	insurance_agency	31	restaurant
8	car_rental	20	laundry	32	train_station
9	car_repair	21	lawyer	33	travel_agency
10	car_wash	22	library	34	university
11	clothing_store	23	lodging		
12	convenience_store	24	movie_rental		

6.1.2 実際の使用環境の再現

実際に提案システムを使用する場合、ユーザは事前に図 2 の①施設の種類を指定する。この施設の種類の再現するため、収集した To-Do を使用し評価を行う際、アンケート回答者毎に、施設の種類の回答の③解決できる施設の集合とし推定を行う。表 6 の場合、この回答者の施設の種類の {6, 7, 12, 13, 16, 17, 18, 25, 31, 34} の 10 施設となる。

6.1.3 精度の算出方法

評価では 1 件の To-Do について、To-Do 解決可能施設推定によって推定された施設が、その To-Do の③解決できる施設に含まれている場合に正解とし、式 (5) によって精度を算出した。

$$\text{精度} = \frac{t}{T} \dots\dots\dots (5)$$

式 (5) の T はテストデータの全 To-Do 数 (T=166) , t は正解した To-Do の数である。

6.1.4 実験と結果

25名の協力者から収集した 166 件の To-Do を用いて提案システムの To-Do 解決可能施設推定を評価する。精度は教師データの量によって変化することが考えられる。また、同義語を検索し教師データに使用する場合としない場合でも変化することが考えられる。実験では、教師データに使用する Google 検索での検索結果から取得する Web ページを各施設上位 10 件から 50 件まで 10 件毎とし、それぞれ教師データに含まれる特徴語の同義語を検索し学習に使用した場合としない場合 10 パターンの分類器を作成した。それぞれの分類器の推定の精度を図 8 に示す。

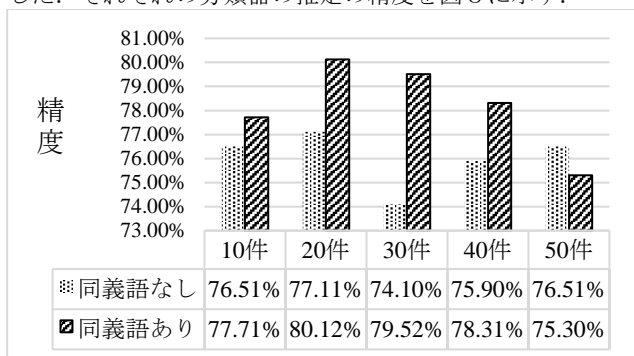


図 8 分類器別推定精度

6.1.5 考察

教師データにユーザが入力する To-Do ではなく、Web ページを使用した図 8 が示すように、教師データに使用する Google 検索での検索結果から取得する Web ページが各施設上位 20 件、教師データに含まれる特徴語の同義語を検索し学習に使用した場合に、80.12% とかなり高い精度でユーザが入力する To-Do の解決可能施設の推定を行うことができた。

Web ページの文章とユーザが入力する To-Do とでは、同じ意味でも違う単語を使用する可能性があるとの理由で同義語を検索し学習に使用したが、これには一定の効果が見られた。図 8 が示すように、教師データに使用する Google 検索での検索結果から取得する Web ページが各施設上位 10 件から 40 件の場合で、最大 5.42% の精度の向上が見られた。しかし、50 件の場合、精度が逆転している。これは教師データの性質が関係していると考えられる。本研究では、教師データは Web ページの文章を使用しており、かつ、その大部分は Google 検索の検索結果から収集している。検索エンジンでは、様々なアルゴリズムを用いて検索結果をソートしており、施設の名称をキーワードとして検索した検索結果の上位のページではその施設に関する特徴語が使われている可能性が高い。しかし、下位に進むにつれてキーワードと関連が薄いページが出現する可能性がある。このページでは他の施設に関する特徴語が使われている可能性がある。そして、その特徴語の同義語を検索し学習に使用することによって、下位のページを学習に使用しない場合と比べてキーワードと関連の薄い施設に推定される可能性が高くなる。これにより、教師データに使用する Google 検索での検索結果から取得する Web ページが各施設上位 50 件では精度が逆転したと考えられる。

教師データはインターネット上で公開されている Web ページの文章、推定に使用したデータは個人が私的に記述する To-Do である。記述した本人が理解すればよい To-Do と不特定多数の人が閲覧する Web ページとでは表記、表現が異なる。推定の精度を向上させるには、教師データに実際にユーザが入力する To-Do を使用する必要があると考える。

以上の結果から、To-Do 解決可能施設推定では、教師データに使用する Google 検索での検索結果から取得する Web ページが各施設上位 20 件、教師データに含まれる特徴語の同義語を検索し学習に使用した分類器を使用する。

6.2 To-Do の通知の評価

提案システムを用いて To-Do が解決できる施設を通知できるか、また複数個所の施設を通知できるかを実験を通して評価する。

6.2.1 実験環境

提案システムは、通勤や通学などの屋外を移動する状況での To-Do の通知を想定している。そこで実験場所を“東京電機大学千葉ニュータウンキャンパス”から最寄り駅の“千葉ニュータウン中央駅”とし、その間にある施設で解決できる To-Do の通知が行われるかを確認する。6.1.1 項のアンケートで収集した 25 名分の回答から 2 人分の回答を選び②To-Do をシステムに入力し、To-Do 解決可能施設推定の結果、正しく推定された (解決可能施設が回答の③

解決できる施設に含まれる場合) To-Do を使用する. 2.2 節の設定画面の項目は以下のように指定する.

- ① 施設の種類…回答の③解決できる施設の集合
- ② 通知間隔…5秒
- ③ 通知範囲…100メートル
- ④ 取得範囲…300メートル

回答者 2 名の回答のうち, 正しく推定された To-Do の概要を表 8 に示す. 表 8 の解決可能施設は表 7 番号-施設対応表を基に変換したものである. 回答者 A は正しく推定された To-Do が 8 件, 推定された解決可能施設は{3,4,6,8,11,27,32,34}であった. 回答者 B は正しく推定された To-Do が 13 件, 推定された解決可能施設は{6,10,14,15,16,19,20,25,26,29,31,32,34}であった.

表 8 各回答者の To-Do の概要

回答者	解決可能施設																	
	3	4	6	8	10	11	14	15	16	19	20	25	26	27	29	31	32	34
A																		
B																		

6.2.2 実験と結果

提案システムを実装したアプリケーションをインストールしたスマートフォン 2 台 (Nexus 5 Android 5.1.1, ZenFone 3 (ASUS_Z012DA) Android 6.0.1) にそれぞれ回答者 A, 回答者 B の To-Do を入力したものを携帯し, 図 9 の甲の地点 (東京電機大学千葉ニュータウンキャンパス) でシステムの動作を開始させ, 太字の線に沿って歩行した. 図 9 の乙の地点 (千葉ニュータウン中央駅) に到着するまでの通知結果を表 9 に示す. 甲から乙までの距離は約 1.15 キロメートル, 徒歩約 15 分である. ○は 1 つの施設, ◎は 2 つの施設が通知されたことを表す. ×は通知範囲内に施設が存在したと考えられるが通知が行われなかったことを表す. - は通知範囲内に施設が存在しないと考えられるため通知が行われなかったことを表す.



図 9 実験経路

表 9 To-Do の通知結果

回答者	解決可能施設																	
	3	4	6	8	10	11	14	15	16	19	20	25	26	27	29	31	32	34
A	-	○	×	-		◎									○		○	○
B			×		-		-	-	-	-	○	-	-	-	○	◎	○	○

6.2.3 考察

実験では, 実験経路付近に存在する 9 種類の施設のうち, 8 種類の施設で通知を確認した. また, 2 種類の施設 (11.clothing_store と 31.restaurant) では, 実験経路付近の 2 つの施設を通知したことを確認した. すべての通知された施設は, 各回答の①To-Do の詳細の内容を実際に解決することができる施設であった. この実験によって To-Do の通知と複数個所の施設の通知が可能であることを確認した. しかし, 6.書店 | 本屋の通知が行われなかった. 実験場所に書店は存在するが, 営業形態が特殊でレンタルビデオ店

と書店が 1 つの建物で営業している. また, この建物に関する情報はレンタルビデオ店のみ API に登録されている可能性がある. このため, 6.書店 | 本屋を通知することができなかったと考えられる. このほか, ショッピングセンターやショッピングモールといった商業施設では, 施設内のすべての店舗の情報が API には登録されておらず, To-Do を解決できるにもかかわらず通知することができない場合がある. また, このような施設だけでなく, API に登録されていない施設について通知を行うことは現状不可能である.

7. まとめ

本稿では, To-Do 入力時のユーザによる To-Do の解決場所の入力なしに, To-Do を解決できる施設を推定し, その施設にユーザが接近した場合に通知を行い, To-Do の解決を支援する “To-Do 解決支援システム” を提案した. ユーザが入力した To-Do が解決できる施設をナイーブベイズ分類器を用いて推定する To-Do 解決可能施設推定, ユーザの周囲にある施設情報を Google Places API を利用し取得する施設情報の取得, To-Do の解決可能施設と施設情報を比較, ユーザの位置情報を基準に To-Do の解決可能施設に接近した際に通知を行う To-Do の通知から構成されている. 25 人の協力者から収集した 166 件の To-Do を用いて To-Do 解決可能施設推定の評価を行い, 教師データに Web ページを使用した場合でも, 実際にユーザが入力する To-Do に対して約 80% の精度で To-Do が解決できる施設の推定が可能であり, かなりの程度有効であることを示した. また, 実際に提案システムを使用する環境を想定した実験を行い, 1 つの To-Do に関して, 1 つの施設のみならずユーザの位置に応じて To-Do を解決できる複数個所の施設の通知が可能であることを確認した.

今後の課題として, テストデータとして使用する To-Do の数を増やし, また To-Do を収集する対象を学生だけでなく社会人へ拡大し評価を行うことで To-Do 解決可能施設推定の有効性をより確かなものにする. 提案システムを実際にユーザが使用する実証実験を行いシステムの有効性を確かなものにするのが挙げられる.

参考文献

- [1] C. Y. Lin, and M. T. Hung, “A location-based personal task reminder for mobile users”, *Personal and Ubiquitous Computing*, vol.18, no.2, pp.303-314 (2014).
- [2] 豊吉 政彦, 後藤 充裕, 錦織 達也, 森脇 康介, 中村 浩司, 木全 英明, “ウェアラブルデバイスとスケジュール解析を用いたコンテキストウェアな行動支援システムの提案”, *電子情報通信学会技術研究報告*, Vol.114, No.479, pp.15-20 (2015).
- [3] 片岡 準, 大瀧 美香, 渡邊 貴之, “位置情報ライフログデータを活用したスケジュール管理支援システム”, *情報処理学会第 75 回全国大会論文集*, pp.285-286 (2013).
- [4] 佐藤 敏紀, 橋本 泰一, 奥村 学, “単語分ち書き用辞書生成システム NEologd の運用 - 文書分類を例にして -”, *自然言語処理研究会研究報告*, Vol.2016-NL-229, No.15, pp.1-14 (2016).
- [5] 田中 幹衛, 吉田 慶介, 松野 省吾, 大山 実, “屋内位置情報を用いた To-Do リストサポートシステムの提案”, *2016 年電子情報通信学会総合大会講演論文集 no.D-9-21* (2016).
- [6] Isahara, H., Bond, F., Uchimoto, K., Utiyama, M., and Kanzaki, K., “Development of Japanese WordNet.”, *Proceedings of 6th International Conference on Language Resources and Evaluation*, pp.2420-2423 (2008).