修 士 論 文

観光ガイドシステムに必要な知識のWeb文書からの自動獲得

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

柿澤 康範

2009年3月

修 士 論 文

観光ガイドシステムに必要な知識のWeb文書からの自動獲得

指導教官　東条敏 教授

審査委員主査　東条敏 教授
審査委員　島津明 教授
審査委員　白井清昭 准教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

710017 柿澤 康範

提出年月: 2009 年 2 月

概 要

対象物が持つ属性情報やトラブル情報を，Web 文書の大規模コーパスを基に自動獲得する研究がこれまでに行われてきた．ユーザがある対象物に関する情報を知りたいといったときに，この自動獲得された知識の一覧を提示すればユーザにとって有用な情報源となるが，ユーザにとって必要な情報を選別して提供できれば更に有用である．

本論文では，ユーザに情報を提供するシステムとして観光ガイドシステムを想定し，Web 文書の大規模コーパスから自動獲得した知識（対象物の属性情報，トラブル情報）を，関連の深い行為を表す動詞や重要度によって分類することで，観光ガイドシステムを利用するユーザが取りたい行動（「行く」や「見る」など）に合わせた情報の提供や，重大なトラブルを優先的に知らせることができるようにすることを目指す．そのために，ユーザのとる行為を表す動詞による属性情報の分類，トラブルによって引き起こされる事象を表す動詞（トラブル動詞）によるトラブル名詞の分類，トラブル動詞の深刻度のランク付けを行った．その結果，トラブル名詞の分類では精度が約 84%，トラブル動詞の深刻度は機械学習による 5 分類の一対比較の精度が約 68%（特定の条件での 2 分類では約 97%）となった．属性情報の分類は約 42%の精度だったが，提案手法はベースラインの手法を上回った．

来年度には，本研究で獲得した属性情報とトラブル情報の知識を，実世界の音声対話システムに組み込む計画を立てている．

キーワード　属性情報，トラブル，Web 文書，大規模コーパス

Abstract

In this thesis we describe automatic classification methods for attribute-value and trouble information on a given topic. The classification methods were designed to cater to users' needs in sightseeing, and the resulting knowledge is to be incorporated in spoken dialog systems of electronic sightseeing guides in Kyoto. More specifically, the goal of this paper is to associate a user's intended action ("go","see", etc.) in sightseeing with particular types of information presented in the form of attribute-value pairs and troubles that are automatically acquired from a huge document collection on the Web.

We attempted 1) to classify attributes according to a user's action such that the action presupposes the user's knowledge of the values of certain attributes and 2) to classify nouns expressing troubles according to their severity, represented as a ranked list of verbs typically associated with those troubles. Using this classification of troubles, a dialog system may select information concerning a relatively small number of specific troubles likely to interfere with particular actions of sightseers from a list of many other troubles.

Experimental results showed 1) that the accuracy of the resulting associations between attributes and actions was around 42%, and 2) that the classification of trouble nouns achieved about 84% accuracy. We also tried to judge the severity of troubles by automatically deciding which one of two given trouble nouns is more serious. The accuracy of this judgement was 68% (with 2-class classification around 97%).

In the next year we plan to use the acquired knowledge on attribute-values and troubles in a real-world spoken dialog system.

Keywords attribute-value, trouble, web document, large corpora

目 次

第1章　はじめに

1.1　研究の目的と背景

まず初めに，本研究で最終的に目標とする，観光ガイドシステムの形態について述べる．図1.1にユーザとシステムのやり取りの一例を示す．１行目でユーザが何をしたいかを述べ，２～４行目でその行動に必要な情報，想定すべきトラブルについて返答している．ユーザからの入力文は，対象物となる名詞と，行動を示す動詞に分けて分析される．この例だと,「清水寺」という対象物に対し,「行く」という行動が示されている．これにより，システムは「清水寺」に関する情報の中から,「行く」に関わる情報である「行き方」を返答する．また，寺に入る際に必須の情報である「拝観時間」と「拝観料」についても返答している．更に，この行動をとる際に想定されるトラブルについて４行目で述べている．５行目はユーザが「金閣寺」を「見る」という入力文であり，それに対する返答として，６行目で「見る」に関わる情報である「見所」を示し，７行目で必須の情報の「拝観時間」と「拝観料」を示している．そして８行目で,「見る」ときに想定されるトラブルとして人混みで疲れる可能性について述べている．

本研究では，このようにユーザが何をしたいことに応じて，対象物に関する情報の中から適切なものを選び，提示することを目指す．なお，このシステムで扱う情報は，対象物が持つ具体的な情報（「清水寺の拝観時間」,「ディズニーランドの入園料」など）の他に，その対象物を利用するときに障害となる可能性があるトラブル（「寺に行くときの渋滞」など）に関する情報も扱う．

ユーザ　：清水寺に行きたい
システム：京都駅から市バス 206 系統に乗り、「五条坂」で降りて下さい
システム：拝観時間は６：００から１８：００、拝観料は３００円です
システム：バスで行く際には、渋滞で遅れる可能性があります

ユーザ　：金閣寺を見たい
システム：見所は、… です
システム：拝観時間は９：００から１７：００、拝観料は４００円です
システム：混雑時は人混みで疲れてしまう場合があります

図 1.1: 観光ガイドシステムの対話例

ユーザが何か情報を知りたいと思ったとき，インターネット上の検索システムを利用することで情報を収集できる．“清水寺” というクエリを入力すれば,「清水寺」に関連するWeb ページの一覧が得られ，そこから辿っていくことで「清水寺」に関する情報が手に入る．しかし，Google や Yahoo などの検索システムでは，ユーザ自身が知るべき情報を正確に把握している必要がある．例えば,「清水寺に行きたいのだけれど，たしか寺に入るためにはお金が必要だった気がする．いくらだろうか？」という疑問を解決するには,“清水寺　拝観料” というクエリを検索システムに入力すれば答えが返ってくるが,「拝観料」という言葉を知らなくては検索ができない．そもそも,「寺に入るためにはお金が必要」という知識すらなかった場合，ユーザが拝観料について調べることもなく，実際に現地に行ってから事実を知ることになる．

こういった，ユーザの前提知識が不足しているときに適切な情報を提供することを目的としたものとして，鳥澤らによる検索ディレクトリ「鳥式」[1] がある．鳥式は，予め対象物ごとに関連語（対象物と関連の深い語）を保持しておき，ユーザが対象物名をクエリとして入力すると関連語の一覧をグラフィカルに提示する（図 1.2）．例えば「清水寺」と入力すると，清水寺に関連する単語が提示され，更に提示された単語をクリックすると，対象物名と関連語をまとめて検索エンジン（yahoo）に送り，その結果を示す．これにより，ユーザが知らなかった，あるいは意識になかった関連語をクエリとして検索エンジンで調べることができるようになる．なお，鳥式では対象物と関連語の知識データは全てコーパスデータから自動獲得されたものであり，対応する対象物の数は 128 万語にもなる．

鳥式では，関連語は「トラブル」,「方法」,「ツール」のカテゴリに分類されている.「トラブル」は対象物を利用する，あるいは対象物に対処する上で障害となる（潜在的）トラブルのカテゴリで，例えば対象物が「ディズニーランド」なら，それを利用する上で障害となる「身長制限」,「渋滞」等がこのカテゴリに属する.「方法」は，対象物を利用／対処する上で有用／必要な具体的方法を含むカテゴリであり，例えばダイエットサプリメントである「ガルシニア」を利用するに当たってはそれを購入する必要があるが，そのための一方法である「輸入代行」などがこれに属する.「ツール」は，対象物を利用／対処する上で用いる道具が属するカテゴリであり，例えば，先ほどのダイエットサプリメントの「ガルシニア」は，対象物が「ダイエット」であった場合はツールのカテゴリで示される．

しかし，鳥式には２つの問題点がある．まず１つ目は，図を見るとわかるように，鳥式では対象物の関連語が一度に大量に表示されるが，その中でユーザが本当に必要とするものは一部だけであり，どれが必要な情報なのか，ユーザ自身が選別する必要があることである．例えば，既にディズニーランドに到着しているユーザが情報を知りたいと思ったとき,「身長制限」というトラブルの情報は役立つが,「渋滞」というトラブルの情報は意味がない．このように，それぞれのユーザの状況に合わせ，ユーザ自身が関連語を選別する必要がある．２つ目としては，鳥式で関連語をクリックして得られるのは関連語に関する具体的な情報ではなくWeb ページの一覧なので，実際の情報は検索エンジンが示すWeb 文書からユーザ自身が見つけ出さなくてはならないということである．例えば,「清水寺」の関連語として「拝観料」が提示されたとしても，それをクリックして得られるのは「拝

観料は○○円」といった情報ではなく，拝観料が書かれている可能性の高い Web 文書の一覧である．

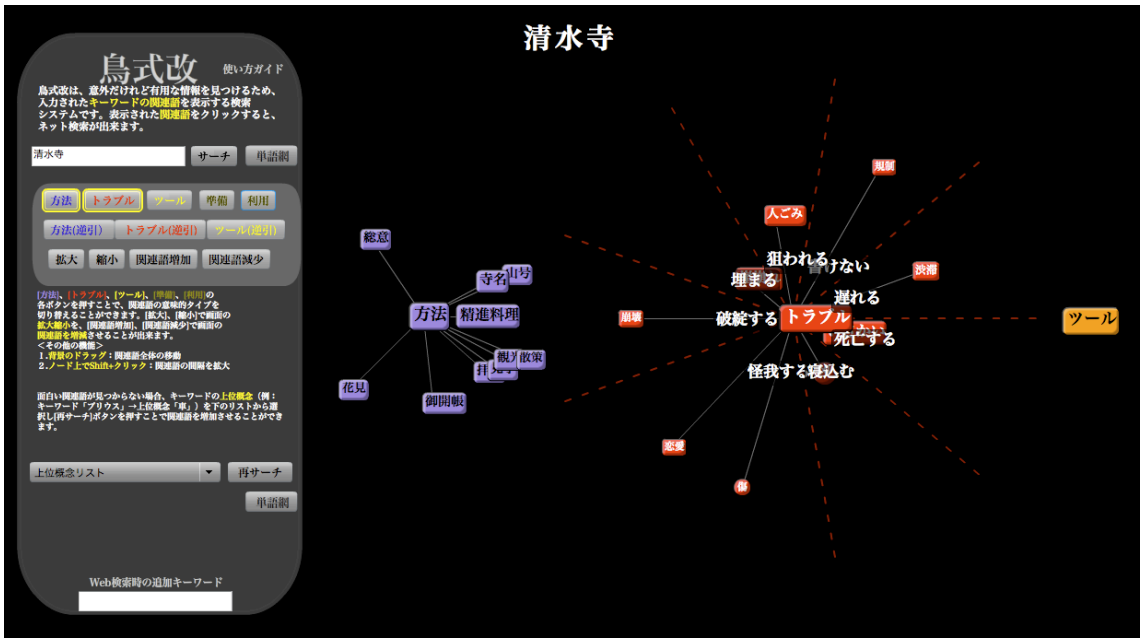

図 1.2: 検索ディレクトリ「鳥式」

鳥式の２つ目の問題点を解決できる研究として，対象物の関連語とそれに対応する情報の組を Web 文書の集合から自動獲得する試みが吉永らによって行われた [3]．吉永らは，Web 文書集合の中から，対象物の**属性**の情報を表や箇条書きなどの視覚的に認知しやすい形で記述したページ（以下，**属性情報記述ページ**）を発見し，属性情報を獲得する研究を行った．ここで属性とは，人が知りたい対象物の側面（例えば寺であれば，「拝観するのにかかる料金」や「寺に行くための方法」）のことであり，文書中では具体的な**属性語**（例：「拝観料」，「交通手段」）によって参照される．これにより，対象物に関する情報を Web 上から収集することが可能となった．しかし，吉永らによって自動獲得された知識は，対象物の情報が一まとまりになったものであり，その中からユーザが必要とする情報を選別しなくてはならない．これは鳥式の１つ目の問題点と同様のものである．

そこで本研究では，Web 文書から自動獲得された知識（属性情報）をユーザがとる行為を表す動詞（「行く」や「見る」など）で分類することで，ユーザが必要とする情報を選別し，「･･･ に行きたい」といったユーザに対しては交通手段や住所などを，「･･･ を見たい」といったユーザには見所，といった状況に合わせた情報提供ができるようにする．更にトラブル情報に関して，トラブルによって引き起こされる事象を表す動詞（「死亡する」，「怪我する」など）でトラブルを分類し，深刻度のランク付けを行うことで，どのような問題を引き起こすトラブルなのかをトラブル名と同時に提示したり，深刻度の大きいトラブルを優先して提示できるようにする．

なお，このような情報提供システムは観光関係に限らず応用可能であるが，本研究では

扱う知識の領域を観光関係に限定する．これは，観光ではユーザのタスクが比較的明確なため，ユーザが必要とする知識を選別する手順，特にユーザの行動プランの推定が行いやすいためである．そのため，本研究では観光ガイドシステムを念頭において，知識の獲得を行う．

本研究では，このようなユーザの取ろうとしている行動に合わせ，適切な情報を提供する観光ガイドシステムを目指し，そのために必要な知識を Web 文書から自動獲得し，知識を分類する．このような観光ガイドシステムを構築するには，対象物の持つ具体的な情報（属性情報，トラブル情報）の他に，ユーザの行動プランを知る必要がある（「寺に行く → バスで行く → バスのトラブルに渋滞がある」）が，本研究ではまず属性情報とトラブル情報の分類を行い，ユーザの行動プランの推定は今後の課題とする．なお，本研究で行うことはユーザに提供する知識の獲得であり，図 1.1 のような自然な対話をどのように行うか，といったことは範囲に含めない．

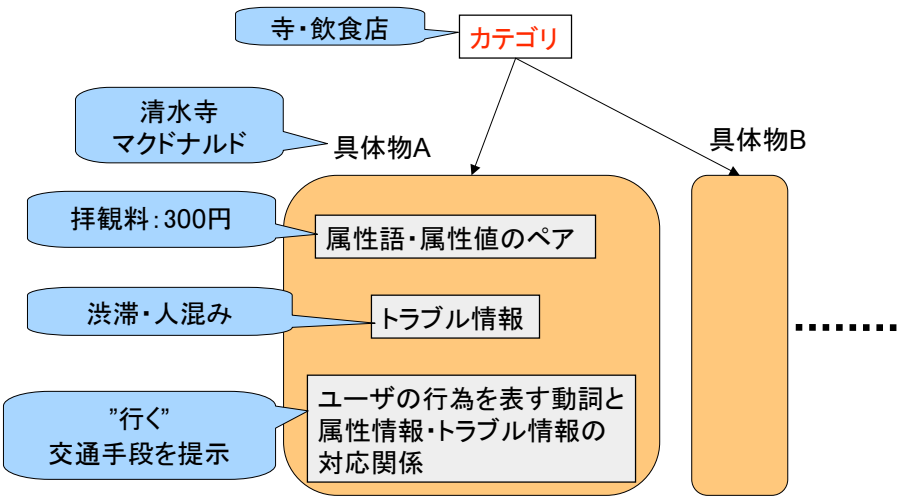

図 1.3: 本研究で自動獲得する知識のデータ構造

このような観光ガイドシステムを実現するために必要な知識を図 1.3 に示す．**カテゴリ**は具体物が属するクラス（上位語）のことであり，それぞれの具体物には，**属性情報**（属性語と属性値のペア），**トラブル情報**，ユーザの行為を表す動詞と属性情報・トラブル情報との対応関係のデータが保持される．これらのデータについては以降の章で解説し，自動獲得を試みる．

1.2 本研究で使用した言語データ

本研究では，新里らによる検索エンジン TSUBAKI[2] で集められた 1 億ページの Web 文書を言語データとして用いた．特に断りがない限り，以降の章の実験で使用されている言語データは，全て TSUBAKI の Web 文書データを基にしている．

1.3 本稿の構成

２章では関連研究として，まず本研究で用いる属性情報・トラブル情報の自動獲得に関する研究の紹介を行い，次にトラブル分類の成果を反映している検索ディレクトリ「鳥式」について紹介する．３章ではユーザのとる行為を表す動詞による属性情報の分類について述べ，４章ではトラブル情報の分類，深刻度のランク付けについて述べる．そして 6 章では本研究の結論と今後の課題について述べる．

第2章　関連研究

本章では関連研究として，観光ガイドシステムでユーザに提供する知識源となる属性情報の自動獲得 [3]，トラブルの自動獲得 [4] について述べる．

2.1　属性情報の自動獲得

本節では吉永らが行った属性情報の自動獲得の概要について述べ，自動獲得の手法の説明を行う．

2.1.1　属性情報の自動獲得の概要

吉永らは，Web 文書集合の中から，対象物の**属性**の情報を表や箇条書きなどの視覚的に認知しやすい形で記述したページ（以下，**属性情報記述ページ**）を発見し，属性情報を獲得する研究を行った．ここで属性とは，人が知りたい対象物の側面（例えば寺であれば，「拝観するのにかかる料金」や「寺に行くための方法」）のことであり，文書中では具体的な**属性語**（例：「拝観料」,「交通手段」）によって参照される．また，各対象物が持つ属性語の具体的な値を**属性値**（例：「300 円」,「○○駅から徒歩×分」）と呼ぶ．属性情報記述ページは，図 2.1 の例のように可読性に優れる上に情報の密度が高く，対象物に関する詳細な情報を効率的に得ることができる．

一方，tf-idf[5] や PageRank[6] などの汎用的なランキング尺度に基づく検索エンジンでは，必ずしも属性情報記述ページが検索結果の上位にくるわけではない．例えば,「清水寺」をクエリとして Google で検索したとき，検索結果の上位にくるページは図 2.2 の例のように，冗長な文章を綴ったページである場合も多く，そこから属性情報を入手するには読解に時間をかけなければならない．

こういった，汎用的な検索エンジンでは得にくい属性情報記述ページを発見し，そのページから属性情報を獲得する手法について，以下の項で述べる．

2.1.2　属性語の獲得

対象物の属性情報記述ページを発見するには，対象物にどのような属性語があるかが重要な手がかりとなると考えられるが，あらゆる対象物について属性語を獲得することは現

図 2.1: クエリ「清水寺」に対する属性情報記述ページの例
URL:http://www.kyotokk.com/kiyomizu.html

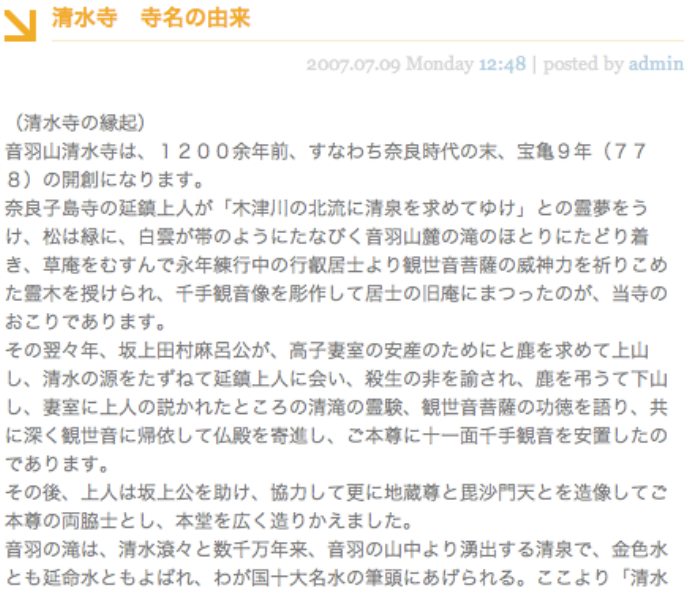

図 2.2: クエリ「清水寺」に対する汎用的な検索エンジンで上位に現れるページの例
URL:http://ishigaki.cc/log/eid807.html

実的ではない．そこで吉永らは，対象物に比べ，文書中により頻繁に出現するクラス（上位語）の単位で属性語を獲得し，属性情報記述ページを発見するための知識源とした．属性語の獲得は，以下の３ステップで行われる．

１．属性情報記述ページの候補となる Web ページの収集　クラスの属性語が多く含まれやすい，クラス名をトピックとした Web ページを集める．具体的には，検索エンジンを用いてクラス名を含む文書を収集し，その中からページのトピックとなる表現が含まれやすい TITLE, H1〜H6, CAPTION, TD[1], および TH タグでクラス名が囲まれているページを抽出し，ページ中でクラス名が最初に現れた位置以降のテキストから属性語候補を獲得する．

２．Web ページからの属性語候補の抽出　属性情報記述ページでは，属性語が HTML タグや文字修飾などによって，視覚的に認知しやすい形で記述されているはずである．そこで，特定の HTML タグまたは括弧類で囲まれた文字列，特定の接頭修飾に続く文字列，および特定の接尾修飾を伴う文字列をパターンにより属性語候補として抽出する．表 2.1 は，吉永らが属性語の抽出に用いた HTML タグと文字修飾である．このようなタグと文字修飾で属性語の候補を獲得できる Web ページの例を図 2.3 に，その Web ページの該当部分の HTML コードを表 2.2 に示す．このページからは，“■” が接頭修飾としてついている「ご案内」や，LI のタグで囲まれている「料金」,「境内自由」,「拝観料」,「宝物館」,「根本堂」,「光明閣・書院庭園」,「拝観時間」,「駐車場」,「住所」,「TEL」,「FAX」が属性語の候補として獲得される．このうち，属性語として適切でない「宝物館」,「根本堂」,「光明閣・書院庭園」といった候補は，後述のフィルタリングで取り除かれる．

HTML タグ: TD, TH, LI, DT, DD, B, STRONG, FONT, SMALL, EM, TT
括弧類: 〔-〕, 【-】, 《-》, ［-］, 〈-〉, ＜-＞, [-], <->
接頭修飾: *, ＊, ●, ○, ■, □, ・, ◆, ◇, ★, ☆, ◎, ・, ○ , ◎
接尾修飾: ：, :, ／, /, ＝

表 2.1: 属性語獲得に用いた HTML タグと文字修飾

３．属性語候補のサイト頻度に基づくフィルタリング　多数の Web ページ製作者が共通して記述する属性語は，ユーザの知りたい典型的な属性語であるという仮説に基づき，以下のように定義される**サイト頻度**が小さい属性語候補は取り除く．

$$sf(x) = \text{属性語候補 } x \text{ を抽出した } \textbf{Web サイトの数} \tag{2.1}$$

ここで言う Web サイトとは，同一 Web ページ製作者が作成した Web ページ群のことである．吉永らは，Web ページの URL（例：`http://ex.org/foo/bar.html`）のパスを末

[1]ただし一行目と一列目のセルに対応するタグのみを考慮する．

■ご案内

- 料金：
  - 境内自由
  - 拝観料
    - 宝物館　300円（要予約、春と秋に一般公開あり）
    - 根本堂　500円
    - 光明閣・書院庭園　600円（抹茶付き）（年末年始は休館）
- 拝観時間：9:00〜17:00（境内は、4〜10月：6：00〜18：00、11〜3月：6：00〜17：00）
- 駐車場：100台
- 住所:島根県安来市清水町528
- TEL:0854-22-2151
- FAX:0854-22-2107

図 2.3: 属性語の候補を獲得できる Web ページの例
URL:http://www.city.yasugi.shimane.jp/p/2/11/4/1/

```
<div><font color="#990000">■ご案内</font></div><ul><li>料金：<ul><li>
境内自由</li><li>拝観料<ul><li>宝物館　300 円（要予約、春と秋に一般公開あり）
</li><li>根本堂　500 円</li><li>光明閣・書院庭園　600 円（抹茶付き）（年末年始
は休館）</li></ul></li></ul></li><li>拝観時間：9:00〜17:00（境内は、4〜10
月：6：00〜18：00、11〜3 月：6：00〜17：00）</li><li>駐車場：100 台</li><li>
住所:島根県安来市清水町 528</li><li>TEL:0854-22-2151</li><li>FAX:0854-22-
2107</li></ul>
```

表 2.2: 属性語の候補を獲得できる Web ページの HTML コードの例

尾から逆に辿り（http://ex.org/foo/→http://ex.org/），Webサイトのトップページのファイル名となりやすい，正規表現/^(?:index|default|main)＼..+/にマッチするファイル名のファイルを含む最下層のディレクトリまでのパスを求め，そのパスをWebサイトと一対一に対応するものと仮定した．ただし，そのようなディレクトリが存在しなかった場合は，サーバー名（例：http://ex.org/）を単にWebサイトとして定義した．

また更なるフィルタリングとして，クラス名をC，属性語をAとしたとき，**「Cの A」**というパターンが一度も現れない属性語Aを候補から取り除いた．

2.1.3 属性語／属性値のペアの獲得

前節で獲得されたクラスの典型的な属性語を用いて，そのクラスに属する対象物の属性語・属性値のペアを獲得する手法は，以下の3ステップからなる．

1．対象物を含むページからの属性語の抽出　対象物名を含むページを検索エンジンを用いて収集し，それぞれのページについて，前節のステップ2で述べた方法を用いて属性語候補を抽出する．

2．クラスの属性知識に基づく属性情報記述ページの発見　ステップ1で抽出された，ページごとの対象物の属性語候補と，対象物が属するクラスの属性語を比較することで，そのページの属性情報記述ページとしての「良さ」を計る．入力の対象物xとそのクラスcに対し，ページpの属性情報記述ページとしての良さを表すスコアを，ページpから獲得した属性語の集合$\mathcal{A}_p$と，前節で述べた方法で獲得されたクラスcの属性語の集合$\mathcal{A}_c$に基づき，以下のように計算する．

$$score(p, c, x) = \frac{\#(\mathcal{A}_p \cap \mathcal{A}_c) \times ratio(\mathcal{A}_p, \mathcal{A}_c)}{ave(\mathcal{A}_p, p) \times text\_size(x, p)} \tag{2.2}$$

ここで，分子の$\#(\mathcal{A}_p \cap \mathcal{A}_c)$は，良い属性情報記述ページはクラスの属性語を多く含むという傾向を反映した項であり，$\mathcal{A}_p$と$\mathcal{A}_c$に共通する属性語の数として計算される．また，$ratio(\mathcal{A}_p, \mathcal{A}_c)$は，対象物が複数のクラスに属する（例：映画とDVDは属する対象物が重なりやすい）場合に，入力のクラスに属する対象物のページを発見するための項であり，$\mathcal{A}_p$に含まれる属性語のうち$\mathcal{A}_c$に含まれる割合（すなわち，$\frac{\#(\mathcal{A}_p \cap \mathcal{A}_c)}{\#(\mathcal{A}_p)}$）として計算される．また分母の$ave(\mathcal{A}_p, p)$は，複数の対象物を含むカタログページよりも，対象物のみについて記述したページを選ぶために用いた項であり，ページp中における全属性語$a \in \mathcal{A}_p$の出現回数（ただし，表2.1のHTMLタグと文字修飾に基づくパターンで抽出されたもののみを考慮する）の平均として計算される．最後に$text_size(x, p)$は，対象物をトピックとして記述するページでは，属性情報のレイアウトに対象物名を含む短い表題が付くことが多いという事実を反映した項である．具体的にこの項は，ページ中で最初に対象物名を含む任意のHTMLタグで囲まれた文字列の長さとして計算される．

このようにして計算された $score(p, c, x)$ が最大のページ p を，クラス c に属する対象物 x の最良の属性情報記述ページとして出力する．

３．属性語／属性値ペアの獲得　ステップ２で得られた属性情報記述ページから，対象物が持つそれぞれの属性語に対応する属性値を抽出する．ここで，与えられた特定の対象物に関する属性語／属性値ペアを獲得する必要があるが，ページ中における対象物名と対象物の属性語／属性値を記述したレイアウトの間の位置関係に関して，吉永らは次のような仮説を立てた．

> **仮説１**
> 与えられた対象物に関する属性語／属性値は，特定の HTML タグで囲まれた範囲（**属性／値ブロック**）に集中して現れる．対象物を記述する属性／値ブロックは，属性語を必ず含み，かつ，そのブロック内，あるいは直前に対象物名を含む．

この仮説に従い，入力の属性語を含むブロックタグ[2]で囲まれた範囲のうち，対象物名を含む，あるいはページ中でその範囲より前の位置に対象物名を含むものを収集し，属性／値ブロックの候補として獲得する．そして獲得された属性／値ブロックについて，以下の仮説に基づき属性／値の記述パターンを導出する．

> **仮説２**
> 属性／値ブロックでは，属性語はその属性値の直前に出現し，更に属性値の直後に別の属性語が続く（属性語-属性名-属性語-属性名- … と続いていく）．属性／値ブロック中では属性は HTML タグや括弧類，接頭・接尾修飾によって強調され，ブロック中の他の属性も同じ強調パターンによって強調される．

具体的には，前節において既に獲得している属性語をページ中から探し，前節での属性語の獲得に用いた HTML タグと文字修飾（表 2.1）のうち，実際にそのページで属性語を強調しているものを抽出する．そしてその強調パターンをそのページ中で探索することにより，属性語の記述の区切りを知ることができ，更にページ中に記述されている未知の属性語も獲得することができる．一方，各属性の値は，対応する属性の直後から，次の属性，あるいはブロック末尾までの文字列として獲得する．

以上の手順で，対象物名とそのクラス名を入力とし，Web ページの集合から属性語／属性値のデータを自動獲得する．吉永らによる実験では，属性語／属性値のペアが正しい事実であると被験者が判断した場合を正解，属性値に正解の事実に加えて無関係の文字列が含まれた場合に準正解とし，611 のオープンドメインの対象物ー属性情報のペアのうち，284(46.5%) ペアが正解もしくは準正解の事実を獲得できた．

[2] title, body, h1, h2, h3, h4, h5, h6, ul, ol, li, pre, dl, dd, dt, div, noscript, blockquote, table, caption, tr, td, th, fieldset, address, p, hr

本研究では，観光に関連したカテゴリに属する対象物について属性語／属性値を獲得し，そのうち属性語について，ユーザのとる行為を表す動詞による分類を行い，ユーザが必要とする属性語の選別を行う．

2.2 トラブルの自動獲得

対象物には，それぞれ特有のトラブルが存在する．例えば,「ディズニーランド」における「順番待ち」や「身長制限」などがある．De Seager らは，トラブルを表す名詞（「渋滞」,「食中毒」など）を自動獲得し，さらに対象物とトラブルの組を自動獲得する研究を行った．本節では，De Seager らが行ったトラブルの自動獲得について述べる．

2.2.1 上位下位関係を利用したトラブル表現の獲得法

トラブルを表す表現（以下，**トラブル表現**）は,「トラブル」という語の下位語といえる．そのため，語彙統語パターンによる下位語の獲得 [7] を利用することができる．図 2.4 は，日本語での下位語の獲得のための語彙統語パターンのリスト [8][9] である．このようなパターンを **LSPH**（ Lexico-Syntactic Patterns for Hyponymy ）と呼ぶ．

トラブルを表す上位語として，De Seager らは「トラブル」,「災難」,「災害」,「障害」を用いた．これらを図 2.4 の＜上位語＞の部分に当てはめ，＜下位語＞の部分を抽出することで，トラブル表現の候補を得ることができる．

1．＜下位語＞　に似た　＜上位語＞
2．＜下位語＞　と呼ばれる　＜上位語＞
3．＜下位語＞　以外の　＜上位語＞
4．＜下位語＞　のような　＜上位語＞
5．＜下位語＞　という　＜上位語＞
6．＜下位語＞　など（の）　＜上位語＞

図 2.4: 下位語の獲得のための日本語の語彙統語パターン

2.2.2 DAV・DNV によるトラブル表現の獲得法

T をトラブル名詞（トラブル表現の名詞），**Y** を対象物とすると，

- T で Y に 行けない
- T で Y が 楽しめなかった

といったパターンで，トラブル名詞と否定形の動詞が同時に現れることが多い．このような，以下の式で表されるパターンを **DNV**（Dependencies to Negated Verbs）と呼ぶ．

$$T \text{ で} \rightarrow \text{否定形の動詞}$$

ただし，DNV だけでトラブル表現を獲得しようとすると適合率が非常に悪く (約 6.5%) なる．これは，例えば「車で○○に行けなかった」といった文が多く現れていれば,「車」がトラブル表現として獲得されてしまうためである．

この問題に対処するため，以下の指標を「トラブル表現ではない度合い」を示すものとして追加する．

$$T \text{ で} \rightarrow \text{肯定形の動詞}$$

このようなパターンを，**DAV**（Dependencies to Affirmative Verbs）と呼ぶ．

2.2.3 トラブルを表す名詞の獲得

トラブルを表す名詞を自動獲得する手順を以下に示す．

1．学習データの収集　まず LSPH や DNV のパターンに当てはまるトラブル表現の候補を集め，以下に示す計算式でスコアを付ける．

$$Score(e) = \frac{f_{LSPH}(e) + f_{DNV}(e)}{f_{LSPH}(e) + f_{DNV}(e) + f_{DAV}(e)} \tag{2.3}$$

ここで $f_{LSPH}(e)$ と $f_{DNV}(e)$，$f_{DAV}(e)$ は，ある表現 e に対し，それぞれ前節で解説したパターンに当てはまった頻度を表している．この $Score(e)$ が大きいほど，トラブル表現である可能性が高い．この後の手順では，ここでのスコアの上位 N 個が用いられる．

2．トラブル表現の発見　SVM（Support Vector Machine）[10] を使った教師あり学習で，トラブル表現と非トラブル表現を分類する．素性には，前節で解説した LSPH，DNV，DAV といったパターンに出現したかどうかの 2 値データと，DNV，DAV における名詞と動詞を結ぶ助詞 (全 5 種類) が共起したかどうかの 2 値データを用いる．2 値データではなく頻度の値を用いても，有意な精度の改善は見られなかった．なお，SVM によってトラブル表現に分類されたものをそのままトラブル表現とするのではなく，正例負例を分割する超平面からの正例側への距離の降順にソートし，その上位 N 個をトラブル表現と見なす．

De Seager らによる実験では，3 人の評価者が全員トラブル表現と判断したものを正解にした場合，適合率 85.5% で 10,000 個のトラブル表現を獲得できた．

2.2.4　対象物とトラブル表現のペアの獲得

対象物と，前節で得られたトラブル表現を関連づけてペアにする手順を以下に示す．

1．対象物とトラブル表現のペアの候補の生成　まず，以下のパターンに当てはまる対象物とトラブル表現のペア $< e_o, e_t >$ を集める．

$$e_o \quad の \quad e_t \tag{2.4}$$

次に，以下の式で示される，pair-wise な相互情報量によってランク付けをし，上位 N 個をペアの候補とする．

$$I(e_o, e_t) = \frac{f(\text{“}e_o \quad の \quad e_t'')}{f(\text{“}e_o'')f(\text{“}e_t'')} \tag{2.5}$$

ここで，$f(e)$ は表現 e の出現頻度である．

2．対象物とトラブル表現のペアのフィルタリング　以下の仮説に従い，フィルタリングを行う．

> **仮説**
> もしトラブル表現 e_t が対象物 e_o を利用する際のトラブルを表しているならば，
> e_o とよく共起し，e_t と以下に示す関係にある動詞 v が存在する．
>
> $$e_t \quad で \quad \rightarrow \text{否定形の動詞} \tag{2.6}$$

具体的には，各対象物ごとに共起頻度の大きい上位 K 個の動詞を集め，それぞれペアの候補となっているトラブル表現 e_t に対し，助詞 “で” と共に否定形になって出現しているかを調べる．そこで K 個の動詞の中で 1 つも当てはまる動詞が無ければ，その対象物とトラブル表現のペアの候補を破棄する．この処理の結果，残った対象物とトラブル表現のペアを，最終的な出力とする．

De Seager らによる実験では，3 人の評価者が全員トラブル表現のペアと判断したものを正解とした場合，適合率 74%で 6,000 対の対象物とトラブル表現のペアを獲得できた．

第3章　属性語の分類

この章では，属性語をユーザのとる行為を表す動詞（「行く」や「見る」など）で分類する．ここで属性とは，人が知りたい対象物の側面（例えば寺であれば,「拝観するのにかかる料金」や「寺に行くための方法」）のことであり，文書中では具体的な**属性語**（例：「拝観料」,「交通手段」）によって参照される．また，各対象物が持つ属性語の具体的な値を**属性値**（例：「300 円」,「○○駅から徒歩×分」）と呼ぶ．

3.1　解決すべき問題

観光ガイドシステムの対話例を図 3.1 に示す．ここで，属性情報を提供している行を強調している．

ユーザ　：清水寺に行きたい
システム：**京都駅から市バス 206 系統に乗り、「五条坂」で降りて下さい**
システム：**拝観時間は６：００から１８：００、拝観料は３００円です**
システム：バスで行く際には、渋滞で遅れる可能性があります

ユーザ　：金閣寺を見たい
システム：**見所は、… です**
システム：**拝観時間は９：００から１７：００、拝観料は４００円です**
システム：混雑時は人混みで疲れてしまう場合があります

図 3.1: 観光ガイドシステムの対話例

清水寺に関する情報を調べようとしているユーザがいたとき，この対話例のように,「清水寺に行きたい」といったユーザに対して「交通手段」や「住所」といった情報を提示し,「清水寺を見たい」といったユーザに対して「見所」などを提示するためには，それぞれの情報に対して,「行く」や「見る」などのユーザのとる行為を表す動詞で分類しておく必要がある．また，対話例中の「拝観時間」や「拝観料」といった情報は，ユーザのとる行為が「行く」でも「見る」でも変わらずに提示されている．これは，この情報がどの状況でも必須の情報であるためで，対話例のような観光ガイドシステムのためには，どの情報が必須のものなのかも獲得しなくてはならない．本研究では，後者の必須の情報の判定

は今後の課題とし，まずは前者の，ユーザのとる行為に合わせて適切な情報を提示できるように分類することを行う．

システムがユーザに提供する情報として，吉永らの研究で自動獲得法が提案されている属性情報を用いる．そして，属性情報のラベルである属性語に対して，ユーザのとる行為を表す動詞で分類することで，対話例のようにユーザに適切な情報を提供するための知識が得られる．表 3.1 に属性語をユーザのとる行為を表す動詞で分類した例を示す．本章では，このような分類を自動的に行う手法を提案し，手法の評価と考察をする．

属性語	ユーザのとる行為を表す動詞
交通手段	行く
住所	行く
見所	見る
ランチメニュー	食べる
駐車場	行く
宿泊施設	泊まる
観覧料	見る
最寄駅	行く
コースマップ	遊ぶ
時刻表	行く
貸し竿	遊ぶ
収容台数	行く
チェックアウト時刻	泊る
見学所要時間	見る
公園時期	見る
エリア	行く
リフト運行時間	遊ぶ
休憩施設	くつろぐ
周辺名所	見る
アクセス	行く

表 3.1: 属性語の，ユーザのとる行為を表す動詞による分類例

3.2 提案手法

属性語と関わりの深い動詞を獲得するためには，まず属性語と係り受け関係にある動詞を抽出することが考えられる．しかし，こういった手法は対象物とユーザのとる行為を表す動詞のペアを獲得する際には有効だが，属性語とユーザのとる行為を表す動詞のペアを

獲得する際には有効ではない．例えば,「住所」は,「住所に行く」といった表現より,「住所を調べる」,「住所を見る」といった表現の方が多い．そこで，単純に属性語と動詞の係り受け関係を調べるのではなく，属性語が属する対象物と動詞との係り受け関係を調べる．

具体的な手順を以下に示す．

1．属性語が属する対象物の収集　属性語 w_a に対し，以下のパターンに当てはまる対象物 w_o を収集する．

$$w_o \quad の \quad w_a \tag{3.1}$$

これは,「清水寺の住所」,「マクドナルドのメニュー」といったように，属性語とその属性語が属する対象物は,「＜対象物＞の＜属性語＞」というパターンで文書中に現れやすいという仮説に基づく．これにより，各属性語ごとに，属する対象物の候補の集合が得られる．

2．対象物と係り受け関係にある動詞の収集　上記で収集した対象物 w_o に対し，以下のパターンに当てはまる動詞 v を収集する．

$$w_o \quad P \quad \rightarrow v \tag{3.2}$$

ここで P は助詞のことで，“で”，“に”，“を”，“は”，“が”，といった助詞が入る．例えば,「京都駅に行く」,「金閣寺を見る」といったものがパターンに当てはまる．このパターンは，De Seager によるトラブル表現の獲得で述べた，DAV とほぼ同様のものである．

3．属性語と動詞のペアのスコア計算　上記の手順で収集したデータを基に，属性語と動詞のペアのスコアを計算する．計算式は以下のようになる．

$$score(w_a, v) = \sum_{w_o \in S_o} \frac{f(\text{“}w_o \quad の \quad w_a'')f(\text{“}w_o \quad P \quad \rightarrow v'')}{f(v)} \tag{3.3}$$

ここで，S_o は上記の手順で収集した，属性語 w_a が属する対象物の候補の集合であり，$f(\text{“}w_o \quad の \quad w_a'')$ と $f(\text{“}w_o \quad P \quad \rightarrow v'')$ はそれぞれのパターンの出現頻度，$f(v)$ は動詞 v の総出現頻度である．この式 3.3 で得られるスコアに従い，各属性語ごとに，最もスコアが高い動詞を選択する．これによって獲得された属性語と動詞のペアが，ユーザのとる行為を表す動詞による属性語の分類結果となる．

なお，提案手法として式 3.3 を用いた理由は，単純に属性語 w_a と係り受け関係にある動詞の頻度をスコアにするより，属性語 w_a と “w_o の w_a” というパターンで共起する対象物 w_o を考慮し，w_o と係り受け関係にある動詞の頻度をスコアにすることで，精度が向上すると考えたためである．この仮説は，例えば「住所」は「住所に行く」といった表現より「住所を調べる」,「住所を見る」といった表現の方が多いが,「X の住所」というパターンに当てはまる具体物 X は場所を表す名詞であることが多く,「X に行く」という

表現が多く出現する，という筆者の観察によるものである．ただし，この具体物 X（対象物の集合 S_o）が特定のカテゴリに偏っていた場合，この仮説通りにはならない．例えば「遊園地」に偏っていた場合,「行く」より「遊ぶ」のスコアが高くなるかもしれない．この問題については，予備実験でコーパスデータを大まかに観察し，筆者の主観で偏りは少ないと判断した．

3.3 実験

3.3.1 観光に関するカテゴリの属性語の獲得

まず，吉永らによる属性語の自動獲得の手法を用い，属性語を獲得した．ここで，対象物の属するカテゴリとして，観光に関する 50 のカテゴリを選別した．これは以下の手順で得た．

1. 「観光」が含まれる語を上位語に持つ下位語を収集する．上位下位語は隅田らによって獲得されたデータ [12][13][14] を用いた．(例：上位語「観光地」→ 下位語「富士五湖」，上位語「観光施設」→ 下位語「ムーミン牧場」）これにより，観光関連の具体物名の一覧が得られる．

2. 上記で収集した観光関連の具体物が下位語となっている上位語を収集する．このとき，いくつの観光関連の具体物の上位語となっているかをカウントする．(以下，**観光具体物頻度**）

3. 「東京都の観光地」のように頭に連体修飾語が付く上位語は，連体修飾語を除いて「東京都の観光地」→「観光地」とする．このとき重複するものは統合し，観光具体物頻度も統合する．

4. 上記で得られた上位語の中で観光具体物頻度が大きい上位 200 個を選び，更に人手で 50 個に選別する．

このようにして得られた 50 のカテゴリに対し，更に類義語・同義語を追加した．これは，獲得する属性語の数を増やすためである．類義語・同義語のデータは，風間らによって自動獲得されたデータ [11] を用い，更に人手でクリーニングした．これにより，50 のカテゴリに合計 291 個の類義語・同義語を追加できた．このデータの一部を表 3.2 に示す．(全体のデータは付録 A に記載）

次に，これらの 50 のカテゴリに対し，吉永らによる属性語の自動獲得の手法を用いて属性語を獲得した．このとき，それぞれのカテゴリの類義語・同義語もクラスの 1 つと見なして属性語を獲得し，その結果はカテゴリごとに統合した．また，獲得した属性語は３人の作業者によってチェックされ，３人中２人以上が属性語として正しいと判断したものを残した．こうして得られた 50 のカテゴリに属する属性語は，重複を除くと 1939 個に

観光に関するカテゴリ	類義語・同義語
ホテル	宿，旅館，ペンション，民宿，ロッジ，お宿，モーテル，ユースホステル
イベント	催し，イヴェント，展示会，催し物，フェスティバル，催事，行事
レストラン	飲食店，食堂，ファミレス
風景	光景，情景，景色，眺め
寺	寺院，お寺，寺社，社寺，本堂，仏殿，お堂，僧院
遊園地	テーマパーク，パーク，アミューズメントパーク
名所	観光名所，観光スポット，景勝地，見どころ，観光ポイント，名勝，名勝地，観光地

表 3.2: 観光に関するカテゴリとその類義語・同義語の一例

なった．獲得した属性語の一部を表 3.3 に示す．なお，この表では 1 つのカテゴリに数個の属性語だけが記載されているが実際には数十個獲得されている．その一例として,「レストラン」に関する全属性語を付録 C に記載する．

3.3.2 ユーザがとる行為を表す動詞による分類

前節で獲得した，観光に関する 50 のカテゴリに属する属性語 1,939 個に対し，提案手法を用いて，ユーザがとる行為を表す動詞で分類した．なお，ユーザがとる行為を表す動詞は，観光に関するものとして以下の 7 個に限定した．この動詞は筆者の主観で選別したものである．

- 行く
- 見る
- 食べる
- 遊ぶ
- 泊る
- 飲む
- くつろぐ

更に比較対象のベースラインとして以下の式で示されるスコアを用いた分類も行った．

$$score(w_a, v) = \frac{f(\text{“}w_a \quad P \quad \rightarrow v'')}{f(v)} \tag{3.4}$$

カテゴリ名	属性語
文化財	交通案内，電子メール，利用料，所有者（管理者），問い合わせ
劇場	入館料，マップ，上映作品名，開館時間，駐車場
伝統行事	開催地，祭りの内容・交通，市町村名，日付
海水浴場	公共交通機関，シャワー・水道，開催時期，備考
温泉	休業日，住所，入浴料金，アクセス，浴用効果
城	開場時間，築城年，サイト URL，最寄駅，電話番号
展望台	交通機関，入館料，公式 HP，駐車場，開放時間
お土産	商品名，お問い合わせ，賞味期限，保存方法
寺	年中行事，宗派，拝観料，アクセス，参拝時間
運動場	施設内容，広さ，休場日，駐車場，設備，場所
遺跡	調査期間，所有者（管理団体），アクセス，出土遺品
公園	交通，レンタル，休園日，付帯施設，イベント情報
喫茶店	営業時間，TEL，FAX，定休日，駐車場，最寄駅
イベント	開催場所，集合場所，申込先・お問い合わせ
ホテル	客室数，ルームタイプ，電話／FAX， 交通アクセス
博物館	公式 HP，閉館日，交通機関，入館料，電話番号
名産品	商品名，製造元，賞味期限，産地，販売期間，販売価格
祭り	実施時期，交通手段，開催場所，問い合わせ，主催
スキー場	斜面構成，駐車場台数，コース紹介，利用料金等
神社	例祭日，お問い合わせ，宮司名，創建年代，エリア

表 3.3: 獲得した属性語の一例

提案手法である式 3.3 が，属性語 w_a と “w_o　の　w_a” という関係にある対象物 w_o を考慮し，その対象物 w_o と係り受け関係にある動詞の頻度を全て用いていたのに対し，このベースラインの方法では，単純に属性語 w_a と係り受け関係にある動詞 v の頻度を基にスコアを計算している．このベースラインの手法については，提案手法の冒頭において，例を挙げながら（「住所」は,「住所に行く」といった表現より,「住所を調べる」,「住所を見る」といった表現の方が多い），適切な手法ではないと述べた方法と同一のものである．

評価実験として，1,939 個の属性語からランダムに 500 個を選び，３人の作業者によって属性語の分類結果をチェックした．このとき，属性語の分類結果として正しい動詞は，必ずしも上記の７個の動詞のいずれか１つになるとは限らない．そのため，上記の７個の動詞以外の動詞に分類されるのが適切な属性語，上記の７個の動詞中に適切な動詞はあるが１つに絞りきれない属性語は，評価の対象外とした．これにより，500 個の属性語から 265 個が除かれ，235 個が残った．そして３人の作業者のうち２人以上が適切な分類だとしたものを正解とし，自動分類したデータの評価を行った．表 3.4 に正解データの一部を示す．

属性語	ユーザがとる行為を表す動詞
駐車サービス	行く
アクセス	行く
最長滑走距離	遊ぶ
出展対象	見る
チェックイン時刻	泊る
アルコール販売	飲む
創建年代	見る
路線名	行く
開催施設	行く
コースデータ	遊ぶ

表 3.4: ユーザがとる行為を表す動詞による属性語の分類の正解データの一例

表 3.5 に実験結果を示す．この結果を見ると，提案手法では正解率が約 42%であまり高くないが，ベースラインの結果と比較すると 15%ほど向上している．これにより，属性語 w_a と “w_o　の　w_a” という関係にある対象物 w_o を考慮し，その対象物 w_o と係り受け関係にある動詞の頻度情報を全て利用して属性語の分類を行う提案手法が，単純に属性語 w_a と係り受け関係にある動詞 v の頻度を基にスコアを計算するベースラインの手法から，十分な正解率の向上を果たしているといえる．

表 3.6 に提案手法がベースラインより適切に分類できた例を，表 3.7 にベースラインのほうが適切に分類できた例を示す．提案手法がベースラインより適切に分類できた例を見ると,「宿泊案内」や「時刻表」など，実際にその属性語のものに対しては「見る」という行為を行うが，ユーザがどのような行為をとるときに必要な情報か，という観点では「泊

る」や「行く」などが適切な分類において，改善されている．一方，提案手法よりベースラインのほうが適切に分類できた例を見ると,「バス」や「公共交通機関」などは前に「＜地名＞の」というパターンが多く現れることが予測されるが，それが提案手法での「泊る」のスコアを大きくしてしまった原因だと考えられる．また他の誤りも，“X の＜属性語＞” というパターンに当てはまる X が，特定のカテゴリに偏ってしまったことが原因だと考えられる．

	正解数	正解率 (%)
ベースライン	62/235	26
提案手法	99/235	42

表 3.5: ユーザがとる行為を表す動詞による属性語の分類の評価結果

属性語	正解	提案手法	ベースライン
分類	見る	見る	行く
アルコール販売	飲む	飲む	行く
宿泊案内	泊る	泊る	見る
開催場所	行く	行く	見る
時刻表	行く	行く	見る

表 3.6: 提案手法がベースラインより適切に分類できた例

属性語	正解	提案手法	ベースライン
駐車サービス	行く	遊ぶ	行く
バス	行く	泊る	行く
公共交通機関	行く	泊る	行く
ランチ	食べる	行く	食べる
登録年	見る	行く	見る

表 3.7: 提案手法がベースラインより適切に分類できなかった例

第4章　トラブルの分類

この章では,「渋滞」や「人混み」といったトラブルを表す名詞（以下，**トラブル名詞**）を,「死亡する」,「遅れる」といったトラブルが引き起こす事象を表す動詞（以下，**トラブル動詞**）で分類する試みについて述べる．また，トラブル動詞の深刻度を求め，ランク付けを行う．

4.1　解決すべき問題

4.1.1　トラブル動詞による分類

観光ガイドシステムの対話例を図 4.1 に示す．ここで，トラブル情報に関する行は強調している．

ユーザ　：清水寺に行きたい
システム：京都駅から市バス 206 系統に乗り、「五条坂」で降りて下さい
システム：拝観時間は６：００から１８：００、拝観料は３００円です
システム：**バスで行く際には、渋滞で遅れる可能性があります**

ユーザ　：金閣寺を見たい
システム：見所は、… です
システム：拝観時間は９：００から１７：００、拝観料は４００円です
システム：**混雑時は人混みで疲れてしまう場合があります**

図 4.1: 観光ガイドシステムの対話例

この対話例では，トラブル情報として「渋滞」と「人混み」について提示しているが,単に「トラブル：渋滞」などと提示するのではなく,「渋滞で遅れる」,「人混みで疲れる」などの形で情報を提供している．このように，トラブル情報をユーザに提供するとき，単にトラブル名詞を提示するだけでなく，そのトラブルによって何が引き起こされるかを同時に提示できれば，特にユーザが詳しくないようなトラブルがあったときに，理解の助けになると考えられる．例えば,「渋滞」のように誰でも意味のわかるトラブル名詞なら良いが,「白飛び」,「こむら返り」といったトラブル名詞は，それが何を引き起こすものなのか

がわからない人も多い．そこで,「白飛びで撮れない」,「こむら返りで痛む」という形で,トラブルによって引き起こされる事象を表す動詞（以下，**トラブル動詞**）も同時に示すことで，ユーザはそのトラブルがどのようなものなのかを，大まかに知ることができる．

こうした情報を提供するためには，トラブル名詞とトラブル動詞を結びつける必要がある．本研究では，これをトラブル名詞をトラブル動詞に分類するタスクとして考え，Web文書から獲得した名詞と動詞の係り受け関係の頻度データや，人手でチェックした教師データを基に，自動分類を試みる．表 4.1 に，トラブルが分類される一例を示す．最終的にはこのような分類を自動的に行うことを目指す．

トラブル名詞	トラブル動詞
渋滞	遅れる
熱中症	倒れる
中毒	死亡する
吹雪	遭難する
満ち潮	水没する
雨	濡れる
交通事故	死亡する
転倒	怪我する
車両点検	遅れる
身長制限	乗れない
脱水症状	倒れる
増水	溺れる
人混み	疲れる

表 4.1: トラブル名詞の，トラブル動詞による分類例

4.1.2 トラブル動詞の深刻度のランク付け

観光ガイドシステムの対話例を図 4.2 に示す．ここで，トラブルの深刻度の大きさによって提示する情報が変わった行は強調している．

トラブルには，深刻なものとそうでないものがある．トラブルの深刻度がわかれば，ユーザにトラブル情報を提供するときに深刻度の大きいトラブルを優先して提示することなどが可能となる．図 4.1 の対話例と図 4.2 の対話例では，最後の行が異なる．図 4.2 のほうは，トラブルの深刻度を考慮し,「人混み」よりも深刻度が大きいトラブルである「スリ」を優先して提示している．このようなトラブル情報の提供の仕方を可能にするには,トラブルの深刻度を求める必要がある．ここで，トラブルによって引き起こされる現象,すなわち共起するトラブル動詞の深刻度が大きいほど，トラブル自体の深刻度は大きいと推測できる．例えば，トラブル動詞による分類で,「A：死亡する」,「B：怪我する」と

ユーザ　：清水寺に行きたい
システム：京都駅から市バス 206 系統に乗り、「五条坂」で降りて下さい
システム：拝観時間は６：００から１８：００、拝観料は３００円です
システム：バスで行く際には、渋滞で遅れる可能性があります

ユーザ　：金閣寺を見たい
システム：見所は、… です
システム：拝観時間は９：００から１７：００、拝観料は４００円です
システム：**スリに盗まれる場合があります**

図 4.2: 観光ガイドシステムの対話例

いう分類になるトラブルＡとＢがあれば,「死亡する」は「怪我する」よりも深刻なので,トラブルＡのほうが深刻であるといえる.

このような深刻度のランク付けをするためには，トラブル動詞の深刻度をランク付けする必要がある．トラブル動詞の深刻度のランク付けができれば，前節で述べたトラブル動詞による分類結果と合わせ，トラブル名詞の深刻度のランクも容易に求めることができる．表 4.2 に，トラブル動詞の深刻度のランク付けの一例を示す．本研究では，このようなランク付けを自動的に行うことを目指す.

深刻度	トラブル動詞
大きい	死亡する
↑	入院する
	怪我する
	汚れる
↓	遅れる
小さい	疲れる

表 4.2: トラブル動詞の深刻度のランク付けの例

4.2　提案手法

＜トラブル表現＞で＜動詞＞
（例：「交通事故で死亡する」,「風邪で休む」）

というパターンで現れる動詞は，トラブルによって引き起こされる事象を示す動詞（トラブル動詞）であり，トラブルを分類するクラスとして利用できる．本節では，こうしたトラブル動詞をクラスとしたトラブル分類と，トラブル動詞の深刻度のランク付けを行う手法について述べる.

4.2.1 係り受け関係を用いたトラブルの分類

単純なトラブル分類として，上記で示したトラブル動詞の定義パターンをそのまま利用し，パターンの出現頻度の最も大きい動詞を分類結果とすることが考えられる．式で表すと以下のようになる．

$$score_{base}(t, v) = f(\text{“}t\text{で}v\text{”}) \tag{4.1}$$

ここでtはトラブル表現，vはトラブル動詞，$f(\text{“}t\text{で}v\text{”})$は「＜トラブル表現＞で＜トラブル動詞＞」というパターンの出現頻度であり，各トラブル表現tについて，$score_{base}(t, v)$が最大になるトラブル動詞vを選択する．

4.2.2 機械学習によるトラブル動詞の深刻度のランク付け

トラブル動詞（例：死亡する，怪我する）の深刻度のランク付けは，局所的に捉えると，あるトラブル名詞 A と B のどちらがより深刻かを一対比較で判断した結果の集合と考えることができる．本研究では，シェッフェの一対比較法 [15] を用いてトラブルの深刻度をランク付けする．また，一対比較の一部は人手で行い学習データとし，残りは SVM (Support Vector Machine)[10] や最大エントロピー法 (ME) によって学習を行い自動分類を行う．

シェッフェの一対比較法は，表 4.3 に示すような 5 段階の評価を，総当たり的に一対比較で行い，それぞれの対象物について，獲得した評価点の平均値を出す．これにより，総当たりで比較した全ての対象物を順序付けることができる．具体的な手順を以下に示す．

トラブル動詞 B から見た A の評価	点数
とても深刻	-2 点
やや深刻	-1 点
同程度	0 点
やや深刻でない	1 点
まったく深刻でない	2 点

表 4.3: トラブル動詞 A と B の一対比較の評価法

1．学習データに対するシェッフェの一対比較法の実施　N 個のトラブル動詞の中から，学習データとして K 個をランダムに選択し，総当たり的に一対比較を行う．この際の評価は表 4.3 に示すような 5 段階で付ける．

２．機械学習による分類　前項で得られた K 個のトラブル動詞に対する総当たりの一対比較データを用い，SVM，最大エントロピー法による学習を行う．一対比較は比較対象のトラブル動詞と比較基準のトラブル動詞の２つのペアによって行われるが，それぞれのトラブル動詞と共起したトラブル名詞とその頻度を素性として用いる．なお，各トラブル名詞に割り振る番号は，比較対象のトラブル動詞と共起したトラブル名詞と，比較基準のトラブル動詞と共起したトラブル名詞で重複しないように，固有の番号を割り振った．

機械学習した分類器を用いて，N 個全てのトラブル動詞の総当たりのペアに対して分類を行い，表 4.3 のような５段階の評価を得る．そして，それぞれのトラブル動詞について，獲得した評価点数を平均化することで，最終的なスコアを得る．このスコアに従ってソートすると，深刻度のランク付けができる．また，ランクの最上位の深刻度を 1.0，最下位の深刻度を 0.0 とし中間のランクのトラブル動詞の値を線形補完することで，簡単ではあるが深刻度の具体的な値を得ることができる．この深刻度の値は，後述するトラブル分類の改善で利用する．

4.2.3　深刻度を用いたトラブル分類の改善

4.2.1 で,「＜トラブル名詞＞で＜トラブル動詞＞」というパターンの頻度を用いた単純な分類法について述べたが，前節で述べた手法で得られるトラブル動詞の深刻度を利用し，分類法の改善を試みる．これは以下の仮説に基づくものである．

> **仮説**
> トラブルによって引き起こされる事象を表す動詞の深刻度は，トラブルと共起する動詞の平均深刻度に近い．

これは例えば,「死亡する」という動詞が分類として適切なトラブル表現があったとすると，このトラブル表現と共起する動詞は「倒れる」,「入院する」,「怪我する」といった比較的深刻なトラブル動詞が多いという筆者の観察によるものである．

以下に，この仮説に基づいてトラブル分類を行う手順について述べる．

１．トラブル名詞の深刻度の計算　各トラブル名詞ごとに共起するトラブル動詞の平均深刻度を求め，これをトラブル名詞の深刻度とする．これは以下の式で示される．

$$SR_t = \sum_{v \in V_t} \frac{SR_v \times f(\text{“}t\text{ で }v\text{”})}{NV_t} \tag{4.2}$$

ここで SR_t はトラブル名詞 t の深刻度，SR_v はトラブル動詞 v の深刻度，V_t はトラブル名詞 t と共起するトラブル動詞の集合，$f(\text{“}t\text{ で }v\text{”})$ は「t で v」というパターンの出現頻度，NV_t はトラブル名詞 t がトラブル動詞と共起した総頻度（$f(\text{“}t\text{ で }v\text{”})$ を全ての v について加算したもの）である．

２．トラブル分類のスコア計算　上記で得られたトラブル名詞の深刻度とトラブル動詞の深刻度を比較し，深刻度の差が小さいペアにより大きなスコアを与える．具体的な式は以下のようになる．

$$score(t, v) = \frac{f(\text{“}t\text{ で }v\text{”})}{1 + \alpha \times |SR_t - SR_v|} \tag{4.3}$$

ここで α は任意の係数である．この式は，式 4.1 の「t で v」というパターンの頻度を単純にとったスコア付けを変形したものであり，トラブル名詞とトラブル動詞の深刻度の差の絶対値が大きいほど，スコアが小さくなる．トラブル名詞をトラブル動詞で分類するときは，トラブル名詞 t について，$score(t, v)$ が最大になるトラブル動詞 v を選択する．

4.3　実験

本研究では 20,183 個のトラブル表現を扱う．このトラブル表現は，Web 文書のコーパスデータから自動獲得 [4] された 30,000 個のトラブル表現を，人手でクリーニングして不適切なものを取り除いたデータである．また，トラブルによって引き起こされる動詞については,「＜トラブル表現＞で＜動詞＞」のパターンに当てはまる頻度が高い動詞を上位 1,000 個抽出し，その中からトラブルを分類するために妥当とされる動詞を人手で 334 個選んだ．更に，これらの動詞の中で類義語，同義語は 1 つのクラスにまとめ，全部で 215 個のクラスとした．表 4.4 と表 4.5 にその一例を示す．

4.3.1　トラブル動詞の深刻度のランク付け

215 のトラブル動詞（のクラス）に対して，深刻度のランク付けを行う．まず，215 個のトラブル動詞からランダムに 50 個を取り出し，１名の作業者によってシェッフェの一対比較法を行った．その比較結果の一部を表 4.6 に示す．ここで，評価値は表 4.3 に基づくもので，比較基準のトラブル動詞（２列目）から見て比較対象のトラブル動詞（１列目）が深刻であれば正の値を，深刻でなければ負の値を，同程度であれば 0 を付ける．また値の絶対値が大きい方がより程度の差が大きい．次に，上記で得られた 50×50 の一対比較結果を学習データ・テストデータとし，１０分割クロスバリデーションで SVM，最大エントロピー法の学習を行った．SVM は工藤が開発した TinySVM[16] を，最大エントロピー法は内山が開発した Maximum Entropy Modeling Package[17] を使用した．素性には，それぞれのトラブル動詞と共起したトラブル名詞と，その共起頻度を用い，SVM ではパラメータを $d = 1$，$C = 1$ とし，カーネル関数には一次の多項式カーネルを使用した（二次の多項式カーネルは予備実験で一次の多項式カーネルより悪い結果になったので，使用しない）．

表 4.7 に SVM による分類結果を，表 4.8 に最大エントロピー法による分類結果をそれぞれ示す．各行はそれぞれ人手での正解データで “-2”，“-1”，“0”，“1”，“2” の評価値に

トラブル表現	
風邪	洪水
騒音	骨折
疲労	副作用
食中毒	渋滞
交通事故	盗難
脱水症状	大気汚染
土砂崩れ	脳梗塞
熱中症	人身事故
豪雨	肌荒れ
山火事	感染症
高血圧	害虫
湿疹	心臓病
地震	ドライアイ
サルモネラ菌	脱線事故
窃盗	日射病
パニック障害	アレルギー疾患
寄生虫	大嵐
光化学スモッグ	踏切事故
速度超過	急性アルコール中毒
残留農薬	寒波

表 4.4: 実験に用いたトラブル表現の一例

トラブル動詞とその類義語・同義語
死亡する，他界する，死去する，急死する
発症する，病む，発病する，患う
倒れる，ダウンする，ぶっ倒れる，倒れられる，たおれる
訴えられる，起訴される，告発される，告訴される，提訴される
怪我する，痛む，傷む，傷つく，痛める，けがする，負傷する
壊れる，破壊される，破壊する，破損する，壊される，こわれる
疲れる，疲弊する，疲れ果てる
眠れない，寝れない
汚れる，汚染される，汚す，汚染する
気絶する，失神する
折れる，折る
止まる，停止する，止められる，ストップする，とまる
動かない，作動しない
負ける，敗れる
歪む，ゆがむ
間違う，間違える，誤る
凍る，凍結する，凍結される
買えない
使えない，使用できない
遅れる，遅刻する

表 4.5: 実験に用いたトラブル動詞の一例

比較対象	比較基準	評価値
死亡する	入院する	-2
怪我する	不足する	-1
自殺する	気絶する	-2
骨折する	逮捕される	2
倒れる	飲めない	0
発症する	水没する	1
感染する	飢える	-1
寝込む	早退する	-1
不足する	水没する	2
迷う	うなされる	1

表 4.6: 人手で行ったシェッフェの一対比較法の結果の一例

なったデータを示し，それぞれのデータについて，４列目から８列目でどの評価値にいくつ分類されたかを示す．そのため，各評価値で正しい分類をされた数は，４列目から８列目と１行目から５行目の，5×5 行列の対角線上の値となる．

	再現率 (%)	適合率 (%)	総数	-2	-1	0	1	2
評価値 -2	70.07	71.90	431	302	112	4	12	1
評価値 -1	64.27	65.42	736	100	473	29	129	5
評価値 0	3.45	5.56	116	7	43	4	59	3
評価値 1	71.47	61.81	736	8	87	32	526	83
評価値 2	67.75	76.04	431	3	8	3	125	292
計	65.18	64.51	2450	420	723	72	851	384

表 4.7: SVM による一対比較法の自動分類結果

	再現率 (%)	適合率 (%)	総数	-2	-1	0	1	2
評価値 -2	67.75	78.07	431	292	126	5	5	3
評価値 -1	72.28	66.25	736	72	532	33	94	5
評価値 0	13.79	17.39	116	2	48	16	48	2
評価値 1	72.55	66.17	736	5	93	33	534	71
評価値 2	67.98	78.34	431	3	4	5	126	293
計	68.04	68.12	2450	374	803	92	807	374

表 4.8: 最大エントロピー法による一対比較法の自動分類結果

この結果を見ると，最大エントロピー法では約68%の精度で，SVM による分類を上回った．また，評価値が “-2” と “2” の一対比較（つまり，評価がはっきりしているペア）において，“-2” が “-1” に分類された場合と “2” が “1” に分類された場合も正解と見なすと（評価値 “0” を考慮しなければ，評価がはっきりとしているペアに限定した２値分類といえる），最大エントロピー法では精度は約97%に達する．なお，表を見るとSVM と最大エントロピー法の両方で，評価値 “0” の結果が非常に悪いが，これは評価値 “0” のペアの総数が116 と，他の評価値のペアに比べて少なかったことが原因だと考えられる．これについては，人手での一対比較データを作成した後，総数の少ない評価値 “0” のペアの総数に合わせ，他の評価値のペアの数を減らすことで解決できるが，全体のデータ量が減ってしまうという問題もある．

215 個の全トラブル動詞に最大エントロピー法による自動分類を行い，その結果得られた各動詞ペアに対する評価値から一対比較法により各動詞の評価値を求めた．その各動詞の評価値が各動詞の深刻度となる．その結果を表 4.9 に示す．（全体のデータは付録 D に記載）ここでは深刻度の上位１０個を示しているが，おおむね正しいランキングになっているように見える．（「去る」が上位にきているが，これは「去る」が人手で作成した学習

データに含まれており，作業者の主観で「去る」が「死ぬ」を言い換える言葉だと判断されたためである）

トラブル動詞	トラブル動詞の評価値
自殺する	-1.99532710
死亡する	-1.96728972
去る	-1.95327103
逮捕される	-1.90186916
水没する	-1.80607477
苦しむ	-1.80140187
訴えられる	-1.79906542
入院する	-1.69158879
感染する	-1.40654206
壊れる	-1.37149533

表 4.9: トラブル動詞の全データのランク付けの結果（上位１０個）

4.3.2 トラブル動詞によるトラブル表現の分類

式 4.1 で定義されるベースラインと，式 4.3 で定義される提案手法を用いて，トラブル表現をトラブル動詞で分類した．提案手法の係数 α は，予備実験で結果の良かった $\alpha = 5$ とした．まず，20,183 個のトラブル表現のうち 3,345 個のトラブル表現をランダムに選んで実験データとし，３人の作業者によってチェックした．そして３人のうち２人以上が正しいとしたトラブル表現／トラブル動詞のペアを正解とした．表 4.10 に正解データの一例を示す．

ベースラインと提案手法による，トラブル表現の分類結果を表 4.11 に示す．この結果を見ると，提案手法はベースラインよりわずかに正解率が低下し，精度の向上は見られなかった．提案手法が精度の向上に結びつかなかった原因としては，まず，深刻度の値の精度が良くなかったことが考えられる．深刻度の値は，本研究でトラブル動詞を深刻度で自動的にランク付けしたデータを用いたが，その時点である程度の誤差があり，特に深刻度が低いトラブル動詞においては誤差が大きかった．しかし，ランキングデータから具体的な深刻度の値を得る際に，単純にランクの最上位を 1.0，最下位を 0.0 として線形補間で値を求めたため，誤差がそのまま残ってしまった（一対比較での評価値をそのまま深刻度とすることも試みたが，予備実験で結果が悪かったので採用しなかった）．改善案としては，ランキングデータから深刻度の値を求める際，単純な線形補間ではなく，ランクの下位の部分では値の変動が小さくなるような補間をすることが考えられる．ただ，トラブル動詞の深刻度の決定の仕方そのものにも問題があり，例えば「怪我をする」といったと

トラブル表現	トラブル動詞
どしゃ降り	遅れる
どしゃ降り	濡れる
どしゃ降り	増水する
転落事故	入院する
転落事故	死亡する
転落事故	怪我する
転落事故	骨折する
吹雪	見えない
吹雪	走れない
吹雪	迷う
吹雪	立ち往生する
熱射病	倒れる
熱射病	入院する
熱射病	死亡する
熱射病	苦しむ
霧	見えない
霧	遅れる
霧	湿る
車両事故	止まる
車両事故	動けない

表 4.10: トラブル動詞によるトラブル表現の分類の正解データの一例

き，どの程度の怪我なのかによって深刻度は大きく変動する．こういった幅のある深刻度をどのように扱うか，といったことも考慮することで改善できるかもしれない．

改善が見られなかった原因としてもう 1 つ考えられることは，単純に式 4.3 のような深刻度を考慮して分類を行う手法に問題がある可能性である．これは，単純にトラブル表現とトラブル動詞の共起頻度を基にするベースラインの手法が分類法として適切であり，そこに深刻度という指標を加えるべきではなかったということも考えられる．これについては，上記の改善案を基に実験を続けていくことで，深刻度という指標が分類に良い影響を与えるのかがわかるものと考えている．

	正解数	正解率 (%)
ベースライン	2,820/3,345	84.30
提案手法	2,802/3,345	83.77

表 4.11: トラブル動詞によるトラブル表現の分類の評価結果

なお，本研究で獲得したトラブル表現の分類データは，鳥澤らによる検索ディレクトリ「鳥式」[1] で使用されている．図 4.3 に鳥式の画面を示す．ここで，中央の「トラブル」という語を中心に，トラブル表現が放射状に広がっているが，類似したトラブル表現は近くになるように配置され，中心からの距離は対象物名（ここでは「ディズニーランド」）との関連度が大きいほど近くなっている．そして，本研究で得られたトラブル動詞による分類結果を基に，それぞれのトラブル表現をまとめている．なお，分類数の少ないトラブル動詞は「その他」でひとまとめにしている．トラブル動詞の並び順については，時計の 3 時方向を基点とし，時計回りに深刻度が大きい順に並んでいる．この深刻度のデータも本研究で獲得したものである．

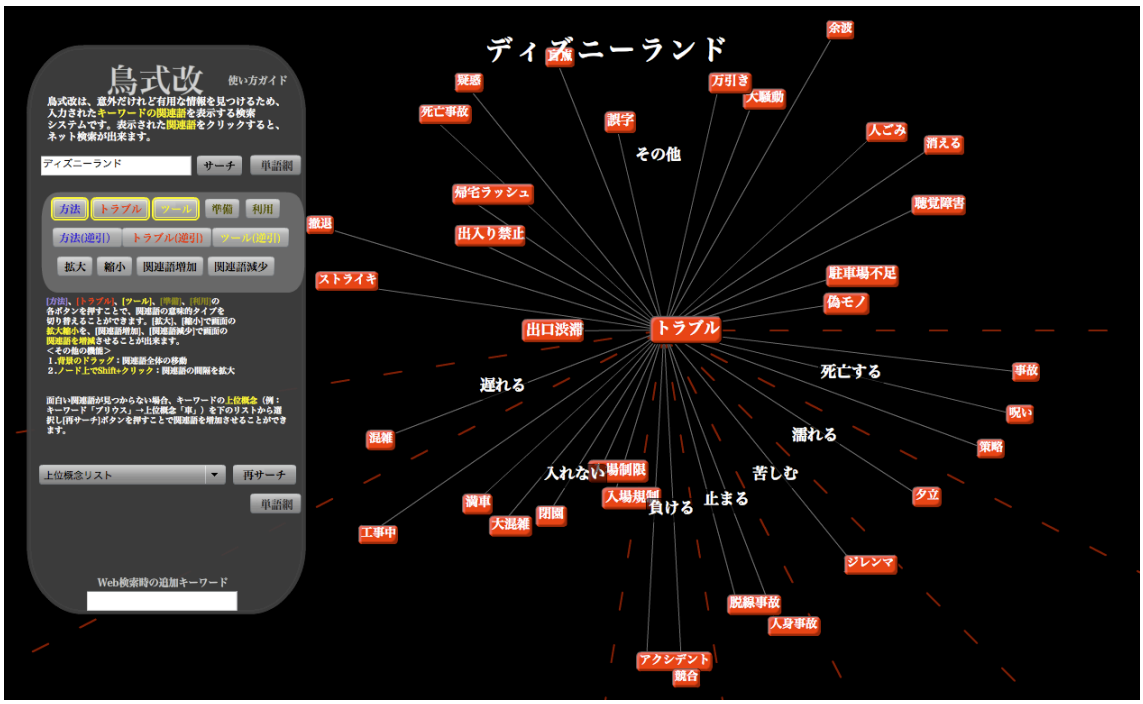

図 4.3: 検索ディレクトリ「鳥式」でのトラブル情報の提示

第5章　おわりに

最後に，本研究のまとめと今後の課題について述べる．

5.1　まとめ

本研究では，観光ガイドシステムに必要な知識を Web 文書のコーパスデータから自動獲得することを目的とし，Web 文書のコーパスデータから得られた知識である，対象物の属性情報とトラブル情報を分類することで，ユーザが必要とする情報を選別できるようにすることを目指した．特にトラブル情報については，深刻度によるランク付けも行った．

属性情報の分類では，属性語をユーザがとる行為を表す動詞で分類した（例：属性語「交通情報」→動詞「行く」，属性語「見所」→動詞「見る」）．その結果，単純に属性語と係り受け関係にある動詞の頻度で分類したベースラインが 26%の正解率だったのに対し，“＜名詞＞の＜属性語＞” というパターンに当てはまる名詞を考慮し，その名詞と係り受け関係にある動詞の頻度で分類した提案手法では 42%となり，15%程度の改善がみられた．ただし，それでも正解率は 50%以下であり，まだまだ改善の余地はあると考えている．

トラブル情報に関しては，まずトラブル動詞（トラブルによって引き起こされる事象を表す動詞,「死亡する」や「遅れる」など）の深刻度を求め，ランク付けした．深刻度のランク付けにはシェッフェの一対比較法を用い，215 個のトラブル動詞のうちランダムに選択した 50 個について，人手で一対比較を行った．そしてその一対比較のデータを学習データとして機械学習することで深刻度をランク付けした．機械学習には最大エントロピー法と SVM を用いたが，最大エントロピー法が再現率，適合率ともに約 65%で，SVM が再現率，適合率がともに約 68%だった．ただし，これは一対比較を５分類（「とても深刻」,「やや深刻」,「同程度」,「やや深刻でない」,「まったく深刻でない」）としたためで，評価がはっきりとしているペア（「とても深刻」,「まったく深刻でない」に人手で分類されたもの）に限定して,「同程度」よりも上か下かの２分類として解釈すると，最大エントロピー法では再現率，適合率ともに約 97%に達する．このように，今回得られた深刻度のランク付けのデータは，評価のはっきりしているトラブル動詞（特に深刻度が大きいもの）についてはかなり良い結果が得られているが，評価のはっきりしないトラブル動詞（特に深刻度が小さいもの）は適切でないランク付けになる傾向にあった．これはそもそも，トラブル動詞は使われる文脈によって深刻度が大きく変化するものが多く，そういったトラブル動詞に対して深刻度を設定するのは，人手でも困難であるという背景がある．

これは，トラブル動詞の深刻度を 1 つの値で表すのではなく，幅のある値として設定することで改善することを考えている．

次に，トラブル表現をトラブル動詞で分類した（例：トラブル表現「食中毒」→ 動詞「入院する」，トラブル表現「渋滞」→ 動詞「遅れる」）．ベースラインの手法では，“＜トラブル表現＞で＜トラブル動詞＞” というパターンの頻度が最も大きいトラブル動詞に分類し，提案手法ではベースラインの手法に「トラブル表現とトラブル動詞の深刻度の差」という指標を加えてスコアを出し，スコアが最も大きいトラブル動詞に分類した．なお，深刻度のデータは本研究で自動獲得したデータを用いた．その結果，ベースラインの正解率は 84.30%，提案手法の正解率は 83.77%となり，正解率の向上は見られなかった．提案手法の結果が悪かった原因としては，深刻度の精度が良くなかったこと，提案手法のような深刻度をトラブル情報の分類の指標にするというアイディアそのものに問題があった可能性，などが考えられる．改善案としては，深刻度の精度を上げること，深刻度以外の指標によるトラブル情報の分類を試みること，などが挙げられる．

以上で，本研究で行った提案手法の実験結果について述べたが，全体として精度がまだ悪く，改善の余地は大きい．今後は，後述する更なる知識獲得と共に，精度の向上も行う必要がある．

5.2 大規模実験の計画

本研究で獲得した知識（属性情報，トラブル情報）は，情報通信研究機構 (NICT) で来年度から本格的に実験が始まる，「京都携帯プロジェクト」に使用される．これは，京都の観光案内を行う多言語対応・音声対応の携帯アプリケーションを作る試みで，コーパスデータから自動獲得した知識を人手でクリーニングしたデータが使用される．提供する予定の知識データを図 5.1 に示す．なお，観光に関する５０のカテゴリに属する，京都と関係の深い対象物（トピックワード）はこれから選別を行うが，プロトタイプとして 200 個のデータは既に選別している．選別はまず，京都府内の駅名 227 個を用い，“＜駅名＞の＜名詞＞” というパターンに当てはまる名詞を抽出し，その名詞の中から５０のカテゴリを上位語に持つ語を選択した．そしてパターンの頻度の大きいものから順に人手でチェックすることで選別を行った．このデータは付録 B に記載する．また，本研究では述べなかった知識データとして，寺や神社などの特定のカテゴリに限定した追加データ（文化財や国宝など）がある．これは wikipedia から項目名を示すパターン “=== 項目名 ===”，サブカテゴリを示すパターン “; 文”，箇条書きのパターン “* 文” などを利用して自動獲得したもので，現時点で 6,200 個程度を獲得している．この追加データの一例を表 5.1 に示す．この表は一例を示したものなので，各対象物につき 2,3 個のデータしか記載されていないが，実際には 1 つの対象物につき数十個のデータを獲得している．

来年度からは，本論文で述べた研究と平行して，「京都携帯プロジェクト」向けのデータ作成も行っていく予定である．

対象物	クラス	データ
大徳寺	国宝	唐門
大徳寺	国宝	方丈および玄関
大徳寺	重要文化財	鳳凰沈金経箱
東福寺	国宝	宋刊本義楚六帖 12 冊
東福寺	重要文化財	絹本著色釈迦三尊像
東福寺	重要文化財	絹本著色応菴和尚像
南禅寺	国宝	秋景冬景山水図
南禅寺	重要文化財	絹本著色仏涅槃図
南禅寺	重要文化財	木造聖観音立像
妙心寺	重要文化財	絹本著色十六羅漢像 16 幅
妙心寺	重要文化財	絹本著色六代祖師像 6 幅
妙心寺	史跡・名勝	方丈庭園
妙心寺	年中行事	1 月 1 日 修二会（しゅにえ）＜正月行事＞
妙心寺	年中行事	2 月 7 日 開山降誕会（ごうたんえ）
醍醐寺	国宝	三宝院表書院
醍醐寺	国宝	絹本著色五大尊像
醍醐寺	重要文化財	遊仙窟
醍醐寺	重要文化財	弘法大師廿五箇条遺告

表 5.1: wikipedia から獲得した文化財・国宝などのデータの一例

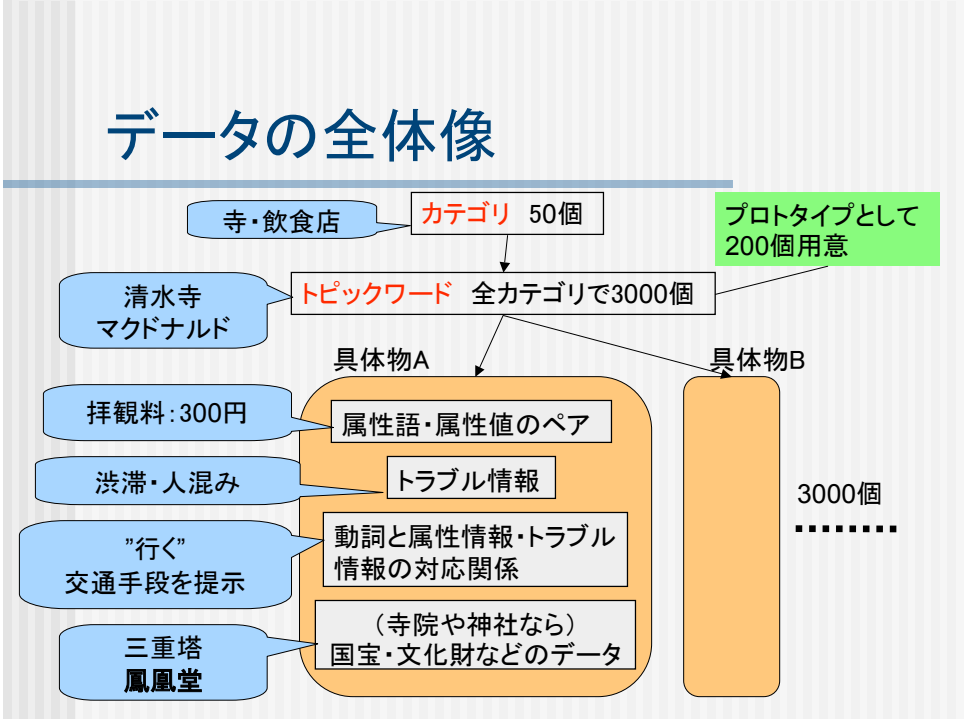

図 5.1: 京都携帯のために提供する知識データ

5.3　今後の課題

まず，本論文で提案した手法はまだ改善の余地があると考えられるため，今後はユーザのとる行為による属性情報の分類，トラブル動詞によるトラブル情報の分類，トラブル動詞の深刻度のランク付けについて，手法の改善を行う．

また，第一章で示したような，目標とする観光ガイドシステムを構築するためには，獲得しなくてはならない知識がまだ多くある．特に，今後の研究の主な内容として，ユーザの行動プランを自動獲得し，情報の選別に利用することを考えている．これは，例えばユーザが「清水寺に行きたい」といったとき，清水寺に行くためには「交通手段」が必要で，もしそれが「バス」ならバスに関連する情報（乗降車するバス停や運賃，時間など），バスに関連するトラブル（渋滞，休日ダイヤ）などを提示する．また更に掘り下げると，「バス」に乗るために○○の行動を取る必要があり…，といった形でユーザが必要とする情報を体系的に提示できるようになる．このようなプランを扱う研究は “プラン認識” という研究分野となるが，プランをコーパスから自動獲得する試みはまだうまく達成できておらず，特に自然言語の文書のコーパスデータから自動獲得することは難しい．主な方針としては，鳥澤によって自動獲得法が提案されている用途／準備表現 [18] や，物事の前後関係を得られるような語彙統語パターンを使い，単純な行動プラン（「清水寺に行く」→「バスで行く」など）から順に獲得をしていきたいと考えている．

謝辞

本研究を進めるにあたり，日頃からたいへんお世話になりました，NICT 言語基盤グループの皆様に深く感謝致します．特に，方針内容に渡って熱心にご指導下さいました，鳥澤健太郎グループリーダーに厚く御礼申し上げます．日頃から研究全般に渡りご指導下さいました，村田真樹主任研究員に深く御礼申し上げます．研究論文や発表資料等をチェックしていただき，様々なご意見を下さいました，風間淳一研究員に深く御礼申し上げます．遠く離れた場所で研究を行う学生の主指導教官を引き受けて下さり，大変お世話になりました，東条敏教授に深く御礼申し上げます．忙しい中，論文審査委員を務めて下さいました，島津明教授，白井清昭准教授に深く御礼申し上げます．属性情報の獲得プログラムに関して，非常に多くのご協力をして下さいました，東京大学の吉永直樹特任助教授に深く御礼申し上げます．トラブル情報のデータや係り受け関係のデータなど，いくつもの言語資源を提供して下さいました，Stijn De Seager 専攻研究員に深く御礼申し上げます．トラブルの深刻度に関する一対比較実験の提案と実験をご協力下さいました，太田公子専攻研究員に深く御礼申し上げます．人手でのデータのチェックに関して，様々なご指導，ご意見を下さり，チェックを行う作業者との仲介をしていただきました，黒田航専攻研究員に深く御礼申し上げます．不十分なチェックマニュアルであったにもかかわらず，データのチェックをして下さり，突発的なデータへの対応もして下さいました，池田千亜紀氏，福岡かおり氏，福森綾子氏に深く御礼申し上げます．

付録

A. 観光に関するカテゴリ一覧

本研究で使用した，観光に関する５０のカテゴリとその類義語・同義語の一覧を表5.2と表5.3に示す.

観光に関するカテゴリ	類義語・同義語
ホテル	宿，旅館，ペンション，民宿，ロッジ，お宿，モーテル，ユースホステル
イベント	催し，イヴェント，展示会，催し物，フェスティバル，催事，行事
レストラン	飲食店，食堂，ファミレス
風景	光景，情景，景色，眺め
寺	寺院，お寺，寺社，社寺，本堂，仏殿，お堂，僧院
遊園地	テーマパーク，パーク，アミューズメントパーク
名所	観光名所，観光スポット，景勝地，見どころ，観光ポイント，名勝，名勝地，観光地
写真	お写真
映画館	シアター
花	お花，華，桜，草花，紅葉
山	山々，山地，山頂，峰
料理	食事，お料理，食べ物，食べもの，飲食，飲食物，食い物，お食事
劇	お芝居，ショー，芝居，オペラ
ビデオ	DVD，VTR，動画，ムービー
島	島々，アイランド
駅	バス停，空港，停留所，タクシー乗り場，乗り場，バスターミナル，ホーム
駐車場	駐輪場，パーキング，自転車置き場，駐車スペース
トイレ	便所，お手洗い，洗面所，おトイレ
公園	広場，庭園，中庭，花畑，お花畑，ガーデン
図書館	図書室
温泉	お風呂，風呂，露天風呂，湯，大浴場，サウナ，銭湯，浴場，おふろ
祭り	お祭り，パレード，祭，まつり，祭典
城	宮殿，砦，お城，王宮，要塞，王城，居城，御殿，本丸
運動場	校庭，グラウンド，グランド，体育館
キャンプ	野宿，キャンプ場，露営，野営，宿営

表 5.2: 観光に関する 50 カテゴリとその類義語・同義語（1）

観光に関するカテゴリ	類義語・同義語
神社	お宮，お宮さん，神宮，大社，社殿
湖・川・海	湖，川，海，沼，渓流，池，入り江， 渓谷，運河，滝，湿原，湾，河
劇場	ホール，ライブハウス，コンサート劇場，シアター
博物館	美術館，資料館，記念館，科学館，水族館，動物園，史料館
甘味処	お菓子屋，菓子屋，ケーキ屋，和菓子屋
喫茶店	コーヒーショップ，カフェ，カフェテリア，喫茶
お土産	おみやげ，土産，みやげ，手土産
飲み物	ドリンク，飲物，ジュース，飲みもの，ソフトドリンク， お飲物，お飲み物，おのみもの
墓	お墓，墓所，墓標，慰霊碑，墓碑，墓場，葬園，霊園
スキー場	ゲレンデ
ゴルフ場	ゴルフコース
釣り場	釣場，釣り座，釣りポイント，フィッシングポイント
居酒屋	飲み屋，パブ，飲み屋さん，バー，スナック， BAR，居酒屋さん
文化財	文化遺産，歴史遺産，歴史的遺産，歴史的建造物，遺物， 文化的遺産，国宝，重要文化財，重文，国指定重要文化財， 国宝重文，国重要文化財，国宝重要文化財，県指定文化財
展望台	見晴台，展望所，展望室，展望デッキ，見晴らし台
ショッピングセンター	ショッピングモール，スーパー，駅ビル，デパート， ファッションビル，モール，スーパーマーケット， ショッピング街，百貨店，ショッピングアーケード，商店
世界遺産	世界文化遺産，文化遺産，自然遺産
ハイキング	トレッキング，散策，山歩き，登山，ピクニック， 行楽，お散歩，山登り
休憩所	待合所，休憩室，レストハウス，東屋，待合室， 休憩場，喫茶室，喫煙所，ラウンジ
海水浴場	ビーチ，海浜公園，浜，浜辺
遊覧船	観光船，水上バス，連絡船，グラスボート，渡し船， 定期船，渡し舟，屋形船，フェリー，高速艇， 汽船，渡船，高速船
案内所	観光案内所，インフォメーションセンター， ツーリストインフォメーション，総合案内所， ツアーデスク，観光センター
伝統行事	伝統芸能，郷土芸能，伝統文化，民俗芸能，民俗行事， 伝統工芸，伝承文化，伝統芸
名産品	特産品，特産物，銘菓，名産物，名産，物産，民芸品， 伝統工芸品，銘酒，地場産品
遺跡	旧跡，史跡，史蹟，古跡，戦跡，遺蹟，古墳

表 5.3: 観光に関する 50 カテゴリとその類義語・同義語（2）

B. 観光に関する対象物のデータ

観光に関する５０のカテゴリに属する具体的な対象物を以下に２００個示す．フォーマットは “＜対象物名＞，＜カテゴリ名＞” となっており，“:” で区切って複数のカテゴリ名が記述されている場合は，複数のカテゴリに属することを意味する．

- あんかけうどん，料理
- いも棒，料理
- お茶，名産品
- き乃ゑ，ホテル
- くいな橋駅，駅
- しだれ桜，文化財
- すぐき，料理
- すす払い，イベント
- たけのこ，名産品
- にしんそば，名産品:料理
- はなふさ，喫茶店
- みなと舞鶴ちゃったまつり，祭り:イベント
- 岩本社，神社
- 顔見世，イベント
- 貴船神社，遺跡:名所:神社:イベント
- 亀岡駅，駅
- 亀岡祭，イベント
- 祇園まつり，祭り:イベント
- 祇園甲部，名所
- 祇園祭，祭り:文化財:イベント
- 祇園祭り，祭り
- 祇園新橋，文化財
- 祇園辻利，甘味処
- 吉原の万灯籠，祭り:イベント
- 吉山明兆，文化財
- 吉富駅，駅
- 久津川駅，駅
- 久美浜シーサイド温泉，温泉
- 久美浜駅，駅
- 宮津駅，駅
- 宮津城，名所:城:イベント
- 牛祭，祭り
- 巨椋池，名所
- 京阪山科駅，駅
- 京漆器，名産品
- 京焼，名産品
- 京田辺駅，駅
- 京都ガーデンホテル，ホテル
- 京都タワー，名所
- 京都パークホテル，ホテル
- 京都ブライトンホテル，ホテル

- 京都駅，駅
- 京都会館，劇場
- 京都祇園祭，イベント
- 京都芸術劇場，劇場
- 京都劇場，劇場
- 京都御苑，公園
- 京都御所，名所
- 京都国際マンガミュージアム，博物館:図書館
- 京都国立近代美術館，博物館:遺跡
- 京都国立博物館，博物館:遺跡:文化財
- 京都三大祭り，祭り
- 京都市考古資料館，博物館
- 京都市図書館，図書館
- 京都市動物園，公園
- 京都市美術館，博物館
- 京都市平安京創生館，博物館
- 京都市役所前駅，駅
- 京都精華大前駅，駅
- 京都全日空ホテル，ホテル
- 京都東急ホテル，ホテル
- 京都府立植物園，名所
- 京都文化博物館，博物館:文化財
- 京野菜，名産品
- 京友禅，名産品
- 京料理，料理
- 恭仁京，遺跡:名所
- 曲水の宴，イベント
- 錦小路通，名所
- 近鉄丹波橋駅，駅
- 金閣寺，名所:文化財
- 金堂，文化財
- 柊家旅館，ホテル
- 百足屋，料理
- 百万遍，イベント
- 普茶料理，料理
- 舞鶴クレインブリッジ，名所
- 舞鶴赤レンガ倉庫群，名所
- 伏見稲荷駅，駅
- 伏見稲荷大社，神社
- 伏見城，名所:城
- 福知山駅，駅
- 平安神宮，遺跡:名所:神社:公園
- 平等院鳳凰堂，文化財
- 米原駅，駅
- 宝ヶ池駅，駅
- 宝ヶ池公園，公園
- 北野天満宮，名所:神社:文化財:イベント
- 北野白梅町駅，駅
- 堀川五条，駅
- 堀川今出川，駅

- 本法寺，寺
- 万願寺とうがらし，名産品
- 妙覚寺，寺
- 妙心寺，文化財:寺
- 明月記，文化財
- 夕日ヶ浦温泉，温泉
- 羅城門跡，遺跡
- 落柿舎，名所
- 嵐山公園，名所:公園
- 瑠璃渓，名所
- 蓮華王院，文化財
- 蓮華寺，遺跡:名所
- 六地蔵駅，駅
- 六波羅蜜寺，文化財:寺
- 和田山駅，駅
- 曼殊院，文化財:寺
- 帷子ノ辻駅，駅
- 涅槃会，イベント
- 聚光院，名所:文化財:寺
- 銀閣寺，名所
- 九条駅，駅
- 桂駅，駅
- 桂川，湖:レストラン:ホテル
- 月読神社，名所:神社
- 建仁寺，遺跡:名所:文化財:寺:イベント
- 五山送り火，イベント
- 五条駅，駅
- 御手洗川，湖
- 光明寺，名所:文化財:寺
- 向島駅，駅
- 向日神社，文化財:神社:
- 広隆寺，名所:文化財:寺
- 行願寺，寺
- 高山寺，名所:文化財:世界遺産:寺
- 高台寺，遺跡:名所:寺
- 高雄山，名所
- 国際会館駅，駅
- 国立京都国際会館，名所
- 今宮神社，神社
- 今出川駅，駅
- 嵯峨野，名所
- 鯖寿司，料理
- 三室戸寺，名所:文化財:寺
- 三十三間堂，遺跡:名所:文化財
- 三条，名所
- 三条駅，駅
- 三条京阪駅，駅
- 三千院，名所:文化財:寺
- 三宝院，文化財
- 山科駅，駅

- 山科神社，神社
- 山城多賀駅，駅
- 山鉾巡行，イベント
- 産寧坂，名所:文化財
- 四条駅，駅
- 四条大宮駅，駅
- 私のしごと館，博物館
- 寺田イモ，名産品
- 寺田駅，駅
- 寺田屋，名所
- 慈照寺，名所:文化財:世界遺産:寺
- 時代祭，祭り
- 七条駅，駅
- 実相院，文化財:寺
- 寂光院，文化財:寺
- 寂光寺，寺
- 蹴上駅，駅
- 出町柳駅，駅
- 上賀茂神社，名所
- 上御霊神社，神社
- 上狛駅，駅
- 上鳥羽口駅，駅
- 城興寺，寺
- 城南宮，名所:神社
- 城陽駅，駅
- 常寂光寺，名所:文化財:寺
- 常念寺，文化財:寺
- 浄瑠璃寺，名所:文化財:寺
- 浄瑠璃寺庭園，名所:文化財
- 新熊野神社，名所:神社
- 真珠庵，文化財
- 真正極楽寺，文化財:寺
- 仁和寺，名所:文化財:世界遺産:寺
- 壬生寺，文化財:寺
- 清水寺，名所:文化財:世界遺産:寺
- 清水寺本堂，文化財
- 精進料理，料理
- 聖護院，文化財:寺
- 西京漬け，料理
- 西陣織，名産品
- 西本願寺，名所:寺
- 石清水八幡宮，文化財:神社
- 千枚漬け，名産品
- 泉涌寺，文化財:寺
- 太秦駅，駅
- 大覚寺，名所:文化財:寺
- 大原，名所
- 大徳寺，名所:文化財:寺
- 大文字五山送り火，イベント
- 醍醐寺，名所:文化財:世界遺産:寺

- 瀧安寺，寺
- 丹後ちりめん，名産品
- 知恩院，名所:文化財:寺
- 地主神社，神社
- 智積院，名所:文化財:寺
- 茶，名産品
- 中尊寺金色堂，文化財
- 長岡天満宮，神社
- 哲学の道，名所
- 天ヶ瀬ダム，名所
- 渡月橋，名所
- 都路里，休憩所:喫茶店:甘味処
- 東映太秦映画村，名所:イベント
- 東山，山:名所
- 東寺，名所:文化財:世界遺産
- 東福寺，名所:文化財:駅:寺
- 湯豆腐，料理
- 湯葉，名産品
- 南禅寺，文化財:寺
- 二条城，名所:文化財:世界遺産:城
- 八坂神社，名所:神社:寺
- 比叡山，山:名所

C.「レストラン」カテゴリの属性語一覧

本研究で獲得した属性語データの中で，一例として「レストラン」カテゴリの属性語の一覧を以下に示す.

- ラストオーダー
- メニュー
- レストラン名
- 駐車場
- 料理
- 価格 (税込)
- カード
- FAX
- アクセス (車で)
- アクセス (電車で)
- TEL・営業時間
- 最寄
- 電話番号
- 店舗名称
- 駐車料金
- 営業
- 店舗名
- 場所
- 店休日
- アクセス
- 席数
- 公式サイト
- 周辺の観光
- 予算
- 収容人数
- 定休日
- カード使用
- 交通
- ジャンル
- 問合せ時間
- 休業
- 駐車台数
- TEL(直通)
- PHONE
- HP
- 概要
- E メール
- ご予約
- 種類
- 駐車場 (収容台数)
- ウェブサイト
- 名前
- 所在地
- 営業時間

- お問い合わせ先
- E-MAIL
- 値段
- ホームページ
- 個室
- 休日
- 価格 (昼)
- 主なメニュー
- FAX(直通)
- 客席
- カテゴリ
- TEL
- HP-URL
- ランチ
- 休業日
- 郵便番号
- 地図
- 住所
- 電話と FAX
- お問合せ先
- TEL/FAX
- 休み
- ディナー
- 問い合わせ先
- 交通アクセス
- 最寄り駅
- 連絡先
- 開店時間
- URL
- 店名
- 料金
- 定休
- 電話
- 時間
- MAIL
- TEL&FAX

D. トラブル動詞の深刻度によるランク付けデータ

本研究で得られた，215 個のトラブル動詞の深刻度によるランク付けデータを以下に示す.

- 自殺する -1.99532710
- 死亡する -1.96728972
- 去る -1.95327103
- 逮捕される -1.90186916
- 水没する -1.80607477
- 苦しむ -1.80140187
- 訴えられる -1.79906542
- 入院する -1.69158879
- 感染する -1.40654206
- 壊れる -1.37149533
- 引退する -1.17289720
- 浸水する -1.10514019
- 失明する -1.09579439
- 自滅する -1.07476636
- 捕まる -1.02570093
- 気絶する -0.99766355
- 冠水する -0.96495327
- 発症する -0.94859813
- 倒れる -0.90654206
- 飢える -0.86448598
- 狂う -0.85280374
- 叩かれる -0.82943925
- 寝込む -0.81775701
- 取られる -0.79906542
- 悲しむ -0.78271028
- 骨折する -0.77336449
- うなされる -0.75233645
- 疲れる -0.74532710
- 怪我する -0.72663551
- 沈む -0.72429907
- 泣く -0.69626168
- 弱る -0.66355140
- 襲われる -0.65887850
- 遅れる -0.64018692
- 沈没する -0.64018692
- 荒れる -0.63785047
- 動けない -0.63551402
- 眠れない -0.63551402
- 切れる -0.63317757
- 決壊する -0.58177570
- 腐る -0.57009346
- 追い出される -0.56775701
- 苦労する -0.55841121
- 折れる -0.55607477

- 失敗する -0.55140187
- 聞こえない -0.54205607
- 止まる -0.53037383
- 被災する -0.52803738
- 不足する -0.50700935
- 停電する -0.50467290
- 汚れる -0.50000000
- 埋まる -0.47196262
- 全壊する -0.47196262
- 休む -0.46962617
- 悪化する -0.42289720
- 働けない -0.39953271
- 麻痺する -0.38785047
- 破綻する -0.38551402
- リタイアする -0.38551402
- 故障する -0.36915888
- 腫れる -0.36448598
- 孤立する -0.35981308
- 焼ける -0.35046729
- 閉鎖される -0.35046729
- 吹っ飛ぶ -0.32009346
- 狙われる -0.31542056
- 増水する -0.31542056
- 揺れる -0.30373832
- 冷える -0.29672897
- 負ける -0.29672897
- 寸断される -0.29439252
- 濡れる -0.28971963
- 削れる -0.28037383
- 中断する -0.27803738
- 歩けない -0.27570093
- 欠ける -0.27570093
- 忘れる -0.25700935
- 休職する -0.24766355
- 動かない -0.23364486
- 嫌われる -0.20794393
- 退場する -0.20794393
- 湿る -0.20327103
- 巻き込まれる -0.20093458
- 消耗する -0.19859813
- 染まる -0.19158879
- 飛び散る -0.19158879
- 割れる -0.18925234
- 吐く -0.18925234
- 使えない -0.17990654
- 取り残される -0.16822430
- 処罰される -0.15186916
- 焼失する -0.13084112
- 溢れる -0.10747664
- 押し寄せる -0.06542056

- 出せない -0.05140187
- 混乱する -0.05140187
- 全焼する -0.02336449
- 来ない 0.00233645
- 立ち往生する 0.00934579
- 漏れる 0.01401869
- 散る 0.01635514
- わからない 0.03738318
- 切断する 0.03738318
- 滑る 0.04672897
- 補導される 0.05373832
- つかない 0.05607477
- 延期される 0.07009346
- 倒産する 0.09345794
- 覆われる 0.09813084
- 頓挫する 0.09813084
- 異なる 0.12149533
- 発熱する 0.14018692
- 歪む 0.14485981
- 揉める 0.14485981
- 曇る 0.14719626
- 生える 0.14953271
- 撤退する 0.15186916
- はまる 0.16355140
- 挫折する 0.17990654
- 閉じ込められる 0.19626168
- 間に合わない 0.19626168
- 乱れる 0.20093458
- ひっくり返る 0.20327103
- 高騰する 0.20560748
- 損する 0.21028037
- 分からない 0.21028037
- 解散する 0.24299065
- 出遅れる 0.25467290
- 抜けない 0.26635514
- 起動しない 0.27336449
- 驚く 0.27570093
- 間違う 0.28504673
- 怒られる 0.29205607
- 溶ける 0.30140187
- 焼け出される 0.30607477
- 逃がす 0.30841121
- 見えない 0.31074766
- 入らない 0.31308411
- 飲めない 0.32943925
- 干される 0.33644860
- 感電する 0.35280374
- 引っかかる 0.36448598
- 搬送される 0.36448598
- 低迷する 0.36915888

- 帰れない 0.37149533
- 焼け落ちる 0.37149533
- 凹む 0.37383178
- 濁る 0.37383178
- 届かない 0.37850467
- 抜ける 0.37850467
- はがれる 0.38785047
- 脱線する 0.44158879
- 鈍る 0.45560748
- 太る 0.45794393
- 立てない 0.45794393
- 凍る 0.47196262
- 変色する 0.48130841
- 広がる 0.49065421
- 固まる 0.50934579
- 浮く 0.50934579
- 食べられない 0.51401869
- 進まない 0.52803738
- 緊張する 0.53271028
- 買えない 0.54205607
- 入れない 0.55607477
- 飽きる 0.56074766
- 規制される 0.57476636
- 乾燥する 0.59345794
- 黒ずむ 0.60514019
- 苦戦する 0.60514019
- 育たない 0.62149533
- かすむ 0.62616822
- 欠席する 0.62850467
- 外れる 0.64485981
- 騒がれる 0.65420561
- 立たない 0.65887850
- 辞める 0.67289720
- 衰える 0.68691589
- 飛ばない 0.69158879
- 欠場する 0.69158879
- 走れない 0.70093458
- 書けない 0.70093458
- 乗れない 0.70093458
- 滞る 0.71028037
- 焦る 0.71962617
- 売れない 0.72897196
- 終らない 0.74299065
- 貼り忘れる 0.75233645
- 黄ばむ 0.76168224
- 迷う 0.77336449
- 合わない 0.77570093
- 販売できない 0.78738318
- 縮む 0.80140187
- 膨らむ 0.80373832

- 傾く 0.81308411
- 読めない 0.83411215
- 繋がらない 0.84579439
- 出られない 0.86448598
- 参加できない 0.87383178
- むくむ 0.88785047
- 伸びる 0.90420561
- 別れる 0.92990654
- 表示されない 0.92990654
- 受けられない 0.93457944
- ずれる 0.93925234
- 劣化する 0.94859813
- 撮れない 0.97196262
- 上がらない 1.03971963
- 曲がる 1.04906542
- 開かない 1.05140187
- 早退する 1.07710280
- 断るれる 1.12616822
- 緩む 1.27803738
- 食べられる 1.46495327
- 鳴る 1.90186916

参考文献

[1] 鳥澤健太郎, 隅田飛鳥, 野口大輔, 風間淳一, ”自動生成された検索ディレクトリ「鳥式」の現状”言語処理学会第 14 回年次大会 pp. 729-732, 2008.

[2] K. Shinzato, T. Shibata, D. Kawahara, C. Hashimoto, and S. Kurohashi. Tsubaki: An open search engine infrastructure for developing new information access. In Proc. of IJCNLP, pages 189-196, 2008.

[3] Naoki Yoshinaga and Kentaro Torisawa, "Open-Domain Attribute-Value Acquisition from Semi-Structured Texts" In Proceedings of the Workshop on Ontolex 2007 – The Lexicon/Ontology Interface held at the sixth International Semantic Web Conference, pp. 55-66. Nov., 2007.

[4] S. De Saeger, K. Torisawa, and J. Kazama. Looking for trouble. In Proc. of The 22nd International Conference on Computational Linguistics (Coling2008), 2008.

[5] Salton, G. and Buckley, C. Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management, Vol. 24, No. 5, pp. 513-523, 1988.

[6] Page, L., Brin, S., Motwani, R., and Winograd, T. The PageRank Citation Ranking. Bringing Order to the Web, Technical report, Stanford Database Libraries Working Paper, 1998.

[7] Hearst, M. Automatic acquisition of hyponyms from large text corpora. In Proc. of COLING'92, pages 539-545, 1992.

[8] Imasumi, K. Automatic acqusition of hyponymy relations from coordinated noun phrases and appositions. Master's thesis, Kyushu Institute of Technology, 2001.

[9] Ando, M., S. Sekine, and S. Ishizaki. Automatic extraction of hyponyms from newspaper using lexicosyntactic patterns. In IPSJ SIG Technical Report 2003-NL-157, pages 77-82. in Japanese, 2003.

[10] Vapnik, Vladimir N. Statistical Learning Theory. Wiley-Interscience, 1998.

[11] Jun'ichi Kazama and Kentaro Torisawa. "Inducing Gazetteers for Named Entity Recognition by Large-scale Clustering of Dependency Relations" In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT) , 2008.

[12] Asuka Sumida, Kentaro Torisawa, and Keiji Shinzato. Concept-instance relation extraction from simple nouns sequences using a search engine on a web repository. In Proc. of the Workshop on the Web Content Mining with Human Language Techonologies, 2006.

[13] Asuka Sumida, Naoki Yoshinaga, and Kentaro Torisawa. Boosting precision and recall of hyponymy relation acquisition from hierarchical layouts in Wikipedia. In Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), 2008.

[14] A. Sumida and K. Torisawa. Hacking Wikipedia for hyponymy relation acquisition. In IJCNLP 2008, 2008.

[15] 佐藤 信, "統計的官能検査法", 株式会社日科技連出版社, 1985.

[16] Taku Kudoh, TinySVM : Support Vector Machines, (http://cl.aist-nara.ac.jp/taku-ku//soft-ware/TinySVM/index.html,2000).

[17] Masao Utiyama, Maximum Entropy Modeling Package, (http://www.nict.go.jp/x/x161/mem-bers/mutiyama/software.html#maxent,2006).

[18] 鳥澤健太郎, "対象の用途と準備を表す表現の自動獲得", 自然言語処理, vol.13, No2, pp125-144, 2006.