確率モデルによる Web ページ推奨エンジン

Web Page Recommendation Engine Based on Probabilistic Modeling

佐藤 健吾 [1)]
Kengo SATO

1) 慶應義塾大学大学院 理工学研究科（〒223–8522 神奈川県横浜市港北区日吉 3–14–1
E-mail: satoken@nak.ics.keio.ac.jp）

ABSTRACT. In this project, a system which recommend web pages for users based on probabilistic modeling was developed. This system assumes that interests of a user are reflected in the history of browsing the pages. The system analyzes the history with probabilistic models which are obtained by Support Vector Machines, so that the system determines recommended pages for the user. The method used in this system could realize a large scale personalization which existing systems can not realize.

1 背景

近年 World Wide Web の爆発的な普及に伴い，各個人の意図や嗜好性にシステムが対応するパーソナライゼーションの重要性が認識されてきている．電子商取引において過去の購入履歴などから顧客の嗜好性を分析し，おすすめの商品を提示するシステムなどの比較的小規模なものが現在では運用されている．このようなシステムは，利用者情報の一括管理とヒューリスティックによる分析で実現されている．利用者情報の一括管理は規模が大きくなると要求されるシステムリソースが膨大になり，ヒューリスティックによる分析では全てのルールを記述することが非常に困難であるという問題点がある．このため，パーソナライゼーションはサイト単位では可能でもインターネットの規模で実現するのは非常に困難であると言える．

一方, Web ページのブラウジングでは, 膨大な量の Web ページの中から目的に合うページを探し出すために検索サイトを利用することが多い．このようなサイトは「ディレクトリ型」と呼ばれるものと「ロボット型」と呼ばれるものに大別することができる．

ディレクトリ型の検索サイトでは，サイト運営者があらかじめ一定の基準で世の中の Web ページを分類しておき，そのリンクをディレクトリのように階層的に配置する．この方法は，人手に頼るために非常に精度は高いが網羅性に欠けるという欠点がある．

ロボット型の検索サイトは，俗にロボットと呼ばれるエージェントプログラムがインターネット上の Web ページのリンクを辿り巡回しながら掻き集めサーバに蓄える．利用者はキーワードを指定し，そのキーワードが含まれる Web ページがサイト独自のスコア順に表示され，その中から目的のページを探す．すべて自動処理で行うため網羅性は高いが，キーワードによる検索しかできないために意図した通りに検索を行うにはかなりの技術が必要である．

このようにいずれの方式も一長一短であり，検索における精度と網羅性はトレードオフの関係にあると言える．

以上のような背景から，本プロジェクトではインターネット規模でも運用することができるパーソナライゼションの手法と，その応用の一つとして Web ページ推奨エンジンの開発を行った．本システムでは利用者の意図や嗜好性は Web ページの閲覧履歴に反映されていると仮定する．これを確率モデルにより分析し，ロボット型検索サイトにおける検索キーワードの代りに用いることで Web ページ推奨エンジンを実現する．

2 目的

本プロジェクトの目的は，インターネット規模でも運用することができるパーソナライゼションの手法の開発である．その有効性を確認するために，応用の一つとして Web ページ推奨エンジンの開発を行った．このようなインターネット規模の推奨エンジンではヒューリスティックによるルール記述は不可能であるため，既存のディレクトリ型検索エンジンから得られる構造をもとに確率モデルの推定を行い，その結果を利用して Web ページの推奨を行う．

3 確率モデルの学習

3.1 サポートベクタマシン

本システムでは, サポートベクタマシン (Support Vector Machine, 以下 SVM) [6] を用いてモデルの学習を行う．この学習法はデータスパースネスに比較的強く，学習データの量が少なくても効率よくモデルを学習することができる．

SVM は，d 個の特徴 (素性) を持つ事例を d 次元の素性ベクトルとして表し，正事例 $\mathcal{X}_1$ と負事例 $\mathcal{X}_2$ に分類されている n 個の学習データから R^d 上の分離平面

$$f(\vec{x}) = \vec{w} \cdot \vec{x} + b \tag{1}$$

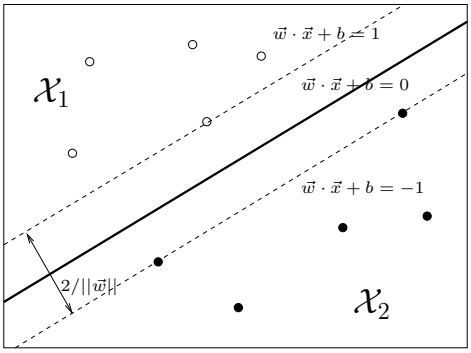


図 1: SVM の分離平面

を学習する．

上述のように線形分離できない学習データに対しては，前処理として非線形な写像を用いてそれらをより高次元に写像することによって線形分離する．

$$\phi : R^d \mapsto R^{d'}$$

このような写像を行い，写像先の空間 $R^{d'}$ で線形分離を行えば元の空間で非線形分離を行っているのと同じことになる．このような写像のために用いる関数はカーネル関数と呼ばれ，SVM の大きな特徴の一つとなっている．

判別したい事例に対応する素性ベクトルを式 (1) に与え，その符号により $\mathcal{X}_1$ に属するか $\mathcal{X}_2$ に属するかを分類することができる．$f(\vec{x})$ の符号がクラスを表し，絶対値が確信度を表す．また，シグモイド関数を用いて $\vec{x}$ が $\mathcal{X}_1$ に属する確率を近似することができる [4].

$$\begin{aligned} P(\vec{x} \in \mathcal{X}_1) &= \tanh(f(\vec{x})) \\ &= \frac{1}{1+\exp(-f(\vec{x}))} \end{aligned} \tag{2}$$

3.2 モデル学習に使用する統計量

SVM でモデルを学習するためには入力となる文書から素性ベクトルを生成する必要がある．本システムでは文書 D に含まれる単語 t の tf-idf 値を素性ベクトルの各要素の値として採用した．

$$\begin{aligned} tfidf(t,D) &= tf(t,D) \times idf(t) \\ tf(t,D) &= t \text{ が文書 } D \text{ で出現した回数} \\ idf(t) &= \log \frac{\text{総文書数}}{t \text{ が出現した文書数}} \end{aligned}$$

文書 D に対して tf-idf 値の値が大きい t はその文書にとって重要であるということを意味する．

文書 D の素性ベクトルは単語 t_i $(i = 1, \ldots, N)$ に対して

$$\vec{x}_D = \begin{pmatrix} \frac{tfidf(t_1,D)}{z} \\ \vdots \\ \frac{tfidf(t_N,D)}{z} \end{pmatrix} \tag{3}$$

と表す．ここで z は $||\vec{x}_D|| = 1$ とするための正規化項である．

HTML 文書において <title>タグ，<a>タグ，<hx>タグに囲まれているテキストは特に重要な意味を持つと考えられるため，これらに含まれる単語を用いて式 (3) と同様にベクトルを作成し，素性ベクトルに加える．したがって素性ベクトルの次元数は $N \times 4$ となる．

3.3 Web ページの特徴量と類似度

既存のディレクトリ型検索サイトのカテゴリを参考に学習データとなる Web ページを人間の手で m 個のカテゴリ C_j $(j = 1, \ldots, m)$ に分類し，それぞれのカテゴリごとにそのカテゴリか否かを識別するモデルを SVM で推定する．その結果，それぞれのカテゴリ C_j に対して分離平面 $f(\vec{x})$ が求まり，式 (2) より $P(D \in C_j)$ を推定することができる．

文書 D は，m 個のカテゴリに対応する確率分布によって m 次元のベクトルによって特徴づけられる．

$$\mathcal{V}(D) = \begin{pmatrix} P(D \in C_1) \\ \vdots \\ P(D \in C_m) \end{pmatrix} \tag{4}$$

文書 D, D' 間の類似度としてベクトルのコサイン距離を用いる．

$$sim(D, D') = \frac{\mathcal{V}(D)}{||\mathcal{V}(D)||} \cdot \frac{\mathcal{V}(D')}{||\mathcal{V}(D')||} \tag{5}$$

4 確率モデルによる Web ページ推奨エンジン

本プロジェクトでは確率モデルによる Web ページ推奨エンジンを開発した．Web ページ推奨エンジンとは，利用者の意図や嗜好性に応じて適切なページを推奨するシステムのことである．本システムは以下のモジュールからなる．

- Model Learning Module
- Crawler Module
- Recommendation Engine
- Personal Proxy Server

これらのモジュールの関係を図 2 に示す．

4.1 Model Learning Module

Open Directory Project (http://dmoz.org/) のカテゴリを用いて学習データを用意し [*1]，3 節で述べた手法により各カテゴリの分類器のパラメータを学習する．パラメータ学習の計算量は膨大であるが，この作業が必要となるのは最初の一回のみである．

4.2 Crawler Module

Crawler Module は，インターネット上の Web ページをリンクを辿り巡回しながら掻き集め，Model Learning Module で学習したパラメータを用いてその特徴を式 (4) に従い特徴ベクトルを計算し，データベースに URL，最終更新日時，文書の特徴ベクトルのみを保存する．HTML 文書自体は保存しないため，ディスク容量は比較的少なくてすむ．

このモジュールは，新しく作られたページ，更新されたページをデータベースに反映させるため，定期的に起動される．

4.3 Recommendation Engine

Recommendation Engine は Personal Proxy Server から送られてきた特徴ベクトルの履歴データと Crawler Module で集められてきた Web ページの特徴ベクトルを類似度 sim (式 (5)) で比較し，値が大きいページの URL を推奨ページとして Personal Proxy Server に送り返す．

Crawler Module でデータベースに蓄えた全てのページとの類似度を計算するのは非現実的であるため，まず履歴データのクラス分類との比較を行い，全てのカテゴリでマッチしているもののみ類似度を計算する．

4.4 Personal Proxy Server

Personal Proxy Server は利用者のコンピュータにインストールされ， Web ブラウザと連携して動作する．Web ブラウザから見ればただの HTTP Proxy として見えるが，内部では利用者が閲覧した Web ページの履歴を特徴ベクトルとして保持する．そして適当なタイミング (Recommendation Engine を配置した Web サーバを閲覧するときなど) にその履歴データを Recommendation Engine に送信し，推奨された Web ページの URL を受け取る．

利用者が同じ嗜好のページをいつまでも見続けることは一般的にはあり得ないため，履歴データの特徴ベクトルも嗜好ごとに分散すると考えられる．このため，履歴データを Deterministic Annealing 法 [3] でいくつかのクラスタにまとめておき，それらの中心値を Recommendation Engine に送信する．

Personal Proxy Server と Recommendation Engine の間の通信は SOAP [1] によるリモートプロシージャコールで行う．

[*1] http://dmoz.org/license.html に示されているライセンスでデータの使用は認められている．

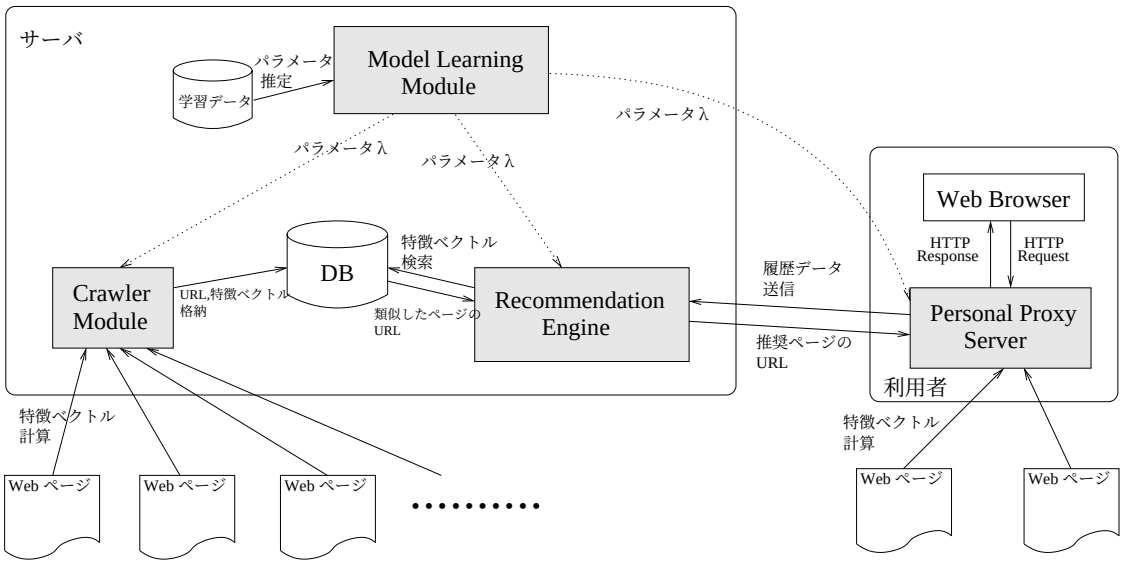


図 2: 本システムの構成

4.5 本システムの特徴

Web ページの閲覧履歴情報を用いてユーザの嗜好性を分析していることが本システムの特徴である．Personal Proxy Server では過去に見た Web ページを事前に推定した確率モデルで分析し，文書ベクトルという形で表す．これにより既存のシステムではできなかった大規模なパーソナライゼーションが可能となる．

また，Recommendation Engine は Personal Proxy Server から文書ベクトルを受け取り，一番似ているページを推奨する．この際，Personal Proxy Server から Recommendation Engine に送信される情報は文書ベクトルのみで，その他の個人情報や実際に見たページの URL などを必要としないため，近年問題となっている個人情報流出は起らず，既存の検索エンジンと同程度のセキュリティである．

通常のキーワード検索はクライアントから送られてきたキーワードを元に検索を行うため，キーワードが多義語だった場合にはユーザが意図しないページが検索結果となる可能性が高い．一方，Recommendation Engine は自身が保持している文書の特徴ベクトルと Personal Proxy Server から送られてきた文書ベクトルを比較する．送られてきた文書ベクトルはユーザの嗜好性が反映されているため，多義語の問題は起こり得ない．

Crawler Module が集めた Web ページは文書ベクトルを求めた後は必要ないので，ページ自体を保存する必要がない．このため，比較的少ないリソースでも本システムを実現することができる．

5 評価

5.1 Machine Learning Module の評価

3 節で述べた手法の有効性を調べるために Web ページ分類の実験を行った．学習データ，テストデータは共に Open Directory Project の日本語のカテゴリ (`http://dmoz.org/World/Japanese/`) より下のカテゴリ群から直接リンクされているページを用いた．素性ベクトルを生成するために KAKASI [*2]により単語を抽出し，出現回数が 5 回以上となる 40,586 語を使用した．したがって素性ベクトルの次元数は $40,586 \times 4 = 162,344$ となった．日本語ページの最上位階層 14 カテゴリについて，含まれる文書のうち半数を学習データ，残りをテストデータの正事例とした．負事例は正事例と同数でその他の 13 カテゴリが同じ割合で含まれるように作成した．以上のように作成した 14 組の学習データで SVM による学習を行い，テストデータでの適合率 (出力された正解数 / 出力数)，再現率 (出力された正解数 / 全体の正解数) を求める実験を行った結果を図 3 に示す．文書数が 500 を越えるカテゴリでは適合率，再現率ともに 7 割以上となり，十分な性能が得られた．一方，文書数が 500 より少ないカテゴリでは再現率が低くなっているカテゴリもある．したがって学習するカテゴリを細かくして学習データの量が少なくなると精度が低くなることが予想される．

5.2 Recommendation Engine の評価

Open Directory Project からリンクされているページのうち，1,903 ページを Recommendation Engine に登録し，それ以外のページから作られる文書の特徴ベクトルを Recommendation Engine に入力し，そのページが属するカテゴリと推奨されたページが属するカテゴリの間の類似度を測ることにより，Recommendation Engine の有効性を確認する実験を行った．カテゴリは木構造になっているので，カテゴリ間の類似度は最上位階層からのカテゴリの一致数を用いる [*3]．SVM 学習用の素性ベクトルは 5.1 節の実験と同様に作成した．文書の特徴ベクトルのもとになるカテゴリ数 (すなわち文書の特徴ベクトルの次元数) として，Open Directory Project の日本語最上位 14 カテゴリを用いた場合と，そのサブカテゴリのうち含まれる文書数が 200 以上の 42 カテゴリを用いた場合の 2 通りの実験を行った．Recommendation Engine に登録されていない 1,903 ページの特徴ベクトルを Recommendation Engine に入力し，類似度 sim が大きい上位 5 個づつを推奨ページとして出力させた際のカテゴリの一致数の分布を図 4 に示す．この図から，次元数 14 よりも次元数 42 で作成した特徴ベクトルを用いた方がカテゴリの一致数が大きいページを多く得られていることがわかる．このことから特徴ベクトルの次元数をより大きくすれば良い推奨ページが得られると考えられる．しかし，学習データの量を変えずに次元数をより大きくするということは 1 カテゴリ辺りの学

[*2] `http://kakasi.namazu.org/`

[*3] 例えば，カテゴリ「スポーツ — ウィンタースポーツ — スキー」とカテゴリ「スポーツ — ウィンタースポーツ — アイスホッケー」の類似度は 2，カテゴリ「スポーツ — ウィンタースポーツ — スキー」とカテゴリ「スポーツ — サッカー — ワールドカップ」の類似度は 1 である．

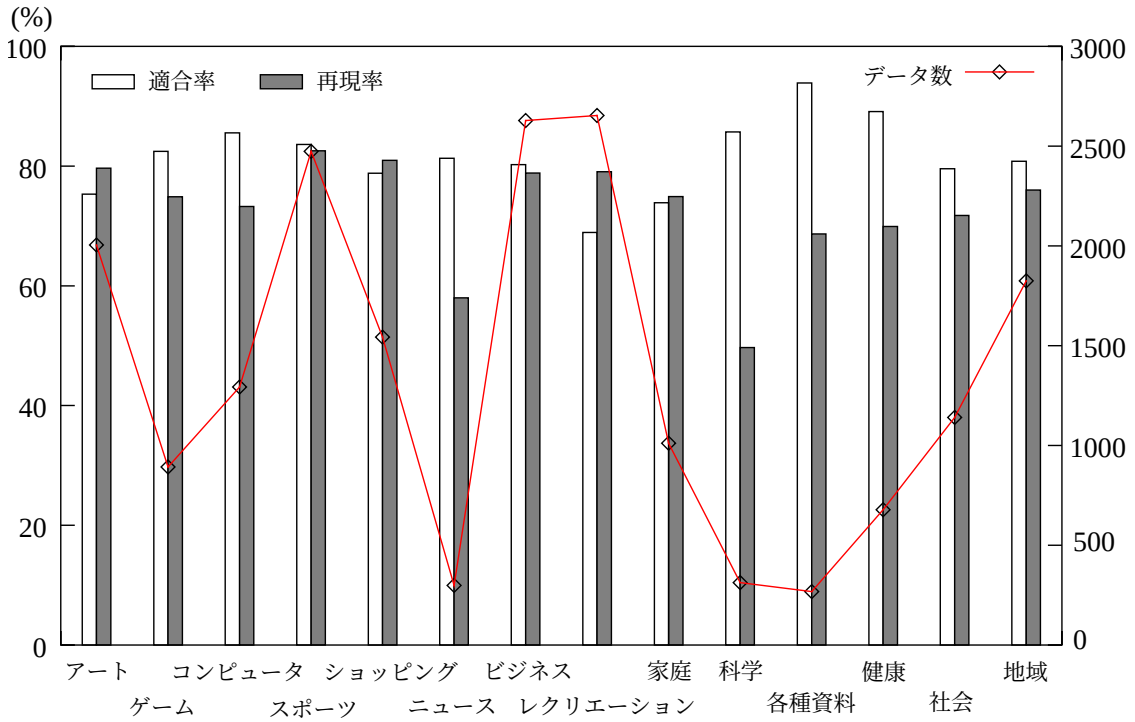
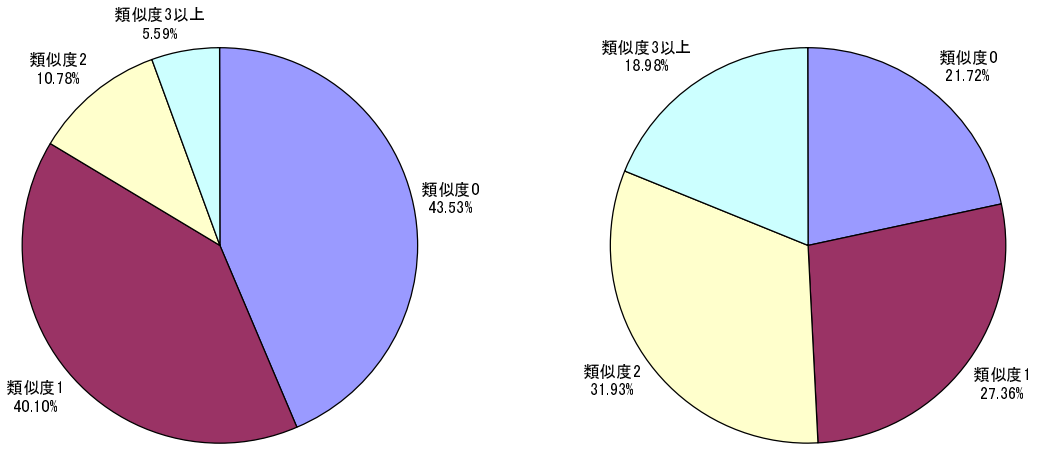


図 3: SVM の分類精度



(a) 次元数 14　　(b) 次元数 42

図 4: Recommendation Engine の精度

習データをより細かく分割することになるため，SVM の学習に対して悪影響を及ぼすことが 5.1 節の結果よりわかっている．これを避けて特徴ベクトルの次元数を増やし Recommendation Engine の性能を向上させるためには学習データ数を増やす他にない．

6　その他の利用シナリオ

本プロジェクトで開発した手法は，Web ページ推奨エンジン以外にも以下のような利用法が考えられる．

検索エンジンの出力結果に対するランク付け

既存のロボット型検索エンジンにおけるキーワード検索は，検索語が含まれる文書をサイトごとの独自のランキング順で表示する．しかし，ここには利用者の意図や嗜好性は反映されない．本プロジェクトで開発した手法を用いて，検索語が含まれる文書と利用者の履歴データの類似度を比較することで利用者の意図や嗜好性を反映したランキングを与えることができる．

閲覧者ごとのページのパーソナライゼーション

Personal Proxy Server を使用しているクライアントからのリクエストに対して Web サーバは履歴データを入手することができるような設定が可能である．履歴データには閲覧者の意図や嗜好性が反映されているので，例えばニュースサイトにおいてスポーツが好きな人に対してはスポーツのニュースをトップ項目に配置するなど，その人の嗜好に合わせてページを動的に生成することが可能となる．

Peer-to-Peer 環境

Personal Proxy Server はユーザの嗜好性を文書ベクトルという形で保持している．Web ページ推奨エンジンではこの情報を Recommendataion Engine で推奨ページを選択するために使用するが，Personal Proxy Server 間でユーザ同士の嗜好性を比較することも可能である．これに

より自分と似た嗜好を持った人を探し，Peer-to-Peer で通信を行うことが可能である．

7 まとめ

本プロジェクトでは確率モデルによる Web ページ推奨エンジンの開発を行った．本システムは，既存のシステムにはない特徴を多く持っている．利用者の意図や嗜好性は Web ページの閲覧履歴に反映されていると仮定し，確率モデルでこれを分析する．これにより既存のシステムではできなかった大規模なパーソナライゼーションが可能となる．

本プロジェクトで開発したシステムの中核部分である Model Learning Module と Recommendation Engine の客観的な評価を行った．今後は実際のサービスと同じ環境でのユーザの主観評価を行う必要がある．また，本プロジェクトで開発したシステムは実験的なものであるため，実際にサービスを提供するためには以下のような問題点がある．

- 確率モデルの学習データの量
  今回学習に使用したデータの量はあまり多くない．これは Open Directory Project に収録されているサイトの数があまり多くないためである．5.1 節で述べたように学習データの量が分類精度を大きく左右することから，質の高い学習データをより多く用意する必要がある．
- 文書ベクトルの次元数，すなわちカテゴリの数
  Open Directory Project に収録されているサイトの数があまり多くないために，細かいカテゴリを使用すると学習データの数が少なくなり精度が低くなる．しかし，5.2 節の実験結果から，文書ベクトルの次元数はユーザの嗜好性を細かく分析するためにはなるべく多い方が良いと思われる．この問題も質の高い学習データをより多く用意することで解決できると思われる．
- Personal Proxy Server のユーザインタフェイス
  今回開発した Personal Proxy Server は実験的なものであるため，ユーザインターフェイスが非常に貧弱である．OS を Windows に，ブラウザを Internet Explorer に限定すれば，Google ツールバー (`http://www.google.co.jp/intl/ja/options.html`) のように使い勝手が良いインターフェイスを実現することが可能であると思われる．

今後は以上のような問題点を解決し，サービスの提供を目指す．

8 参加企業および機関

- 財団法人京都高度技術研究所

謝辞

本プロジェクトの実施にあたり，プロジェクトマネージャの京都大学上林弥彦教授には終始ご指導と貴重な機会を与えて頂いたことに深く感謝致します．また，プロジェクト管理組織の京都高度技術研究所の三好様，杉本様，平家様にはプロジェクトの実施を厚くサポートして頂きました．様々な機会にご意見を頂きました皆様，未踏ソフトウェア創造事業を担当された IPA の関係者各位に深く感謝致します．

参考文献

[1] Simple Object Access Protocol (SOAP) 1.1. `http://www.w3.org/TR/SOAP/`, 2000.

[2] 北研二．確率的言語モデル, 言語と計算, 第 4 巻．東京大学出版局, 1999.

[3] Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 1993.

[4] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[5] 徳永健伸．情報検索と言語処理, 言語と計算, 第 5 巻．東京大学出版局, 1999.

[6] Vladimir Naumovich Vapnik. *The Nature of Statistical Learning Theory (Statistics for Engineering and Information Seience)*. Springer-Verlag Telos, 2nd edition, December 1999.