検索隠し味を用いた専門検索エンジンの構築

小久保 卓[†,☆] 小山 聡[†] 山田晃弘[††,☆☆]
北村泰彦[†††] 石田 亨[†]

いまやインターネットは現代社会の中に急速に浸透しており，そのサービスの中でも特に WWW（World Wide Web）は新しいメディアとしてその情報量を増大させている．しかしながら最も一般的な WWW 情報検索手法である検索エンジンは，必要な情報を得るためにある程度の知識や経験が必要とされ，多くの初心者にとって使いこなすのは容易ではない．こうした WWW 情報検索における問題の解決法の 1 つとして，ドメインを限定した専門検索エンジンの提供があげられている．そこで本論文では専門検索エンジンを構築するための新しい手法として “検索隠し味” を用いた手法を提案する．これはユーザの入力クエリに対しある特定のキーワードを追加すると，汎用検索エンジンの出力のほとんどがドメインに関係する Web ページとなるという経験則を利用したものである．そして機械学習の一種である決定木学習アルゴリズムを元に Web ページ集合からキーワードのブール式の選言標準形として検索隠し味を抽出するアルゴリズムを開発した．さらに本手法を料理レシピ検索に適用し評価実験を行うことで，その有効性の確認を行った．

Keyword Spice Method for Building Domain-specific Web Search Engines

Takashi Kokubo,[†,☆] Satoshi Oyama,[†] Teruhiro Yamada,[††,☆☆]
Yasuhiko Kitamura[†††] and Toru Ishida[†]

The WWW technology has come into wide use in our society as an infrastructure that supports our daily life. But gathering information from the WWW is a difficult task for a novice user even if he uses the search engines that are most widely used tool to retrieve information from the WWW. Because the user must have experience and skill to find the relevant pages from the large number of documents returned, which often cover a wide variety of topics. One solution to the problem is to build a domain-specific search engine. So this paper presents a new method that improves search performance by adding the domain-specific keywords, called *keyword spices*, to the user's input query; the modified query is then forwarded to a general-purpose search engine. We describe a machine learning algorithm, which is a type of decision-tree learning algorithm, that can extract keyword spices as a disjunctive normal form of keywords from Web documents. To demonstrate the value of the keyword spices, we conducted experiments in the cooking domain and the results showed the high performance.

1. はじめに

いまやインターネットは現代社会の中に急速に浸透しており，我々の日常生活を支えるインフラストラクチャの 1 つとなってきている．インターネットが提供するサービスの中でも利用者間での情報共有を実現する WWW（World Wide Web）は最も人気の高いものであり，WWW に蓄積されている情報量は日々，急速な勢いで増加を続けている．しかし最も一般的な WWW 情報検索手法である検索エンジンは，大量の結果から必要な情報を選択するためにある程度の知識や経験が必要とされ，多くの初心者にとって使いこなすのは容易ではない．多くのユーザは多数のキーワー

† 京都大学大学院情報学研究科社会情報学専攻
Department of Social Informatics, Graduate School of Informatics, Kyoto University
†† イメージ情報科学研究所
Laboratories of Image Information Science and Technology
††† 大阪市立大学大学院工学研究科情報工学専攻
Department of Information and Communication Engineering, Graduate School of Engineering, Osaka City University
☆ 現在，NTT ドコモ
Presently with NTT DoCoMo, Inc.
☆☆ 現在，三洋電機株式会社
Presently with SANYO Electric Co., Ltd.

ドからなる詳細な検索式を作成することができず，その結果として大量の無関係のページが検索されてしまう[2]．こうした WWW 情報検索における問題の解決法の 1 つとして，ドメインを限定した専門検索エンジンの提供があげられている[6]．

そこで本論文では専門検索エンジンを構築する新たな手法として，機械学習により Web ページ集合から抽出されたキーワードを用いた手法を提案する．

我々の 1 人が日本語の検索エンジン（goo）を用いて牛肉を使ったレシピを検索していたとき，“牛肉” というキーワードでは結果のトップ 25 件のうち 15 件だけ（60%）がレシピの Web ページであった．そこで新たに “塩” を追加して検索したところ，驚くことに 1 つを除く，24 件の結果（96%）がレシピページとなった．豚肉や鶏肉といった他の食材で試してみたところ，同様の非常に良い検索結果を得ることができた．このことはユーザが入力するクエリにあるキーワードを追加して汎用検索エンジンに転送することで，専門検索エンジンが構築できるという可能性を示していた．この発見を一般化したものが我々の “検索隠し味” を用いた手法である．

これまでに様々な専門検索エンジンの研究が行われているが，最も単純な専門検索エンジン構築手法は，ロボットによりドメインに関係する Web ページだけを収集してインデックス化したものである．例としてコンピュータサイエンスの研究論文のための検索エンジンである Cora☆[6] があげられる．ここでロボットは分野に特化した学習手法により WWW を効率的に探索している．SPIRAL[3] や WebKB[4] も同様にロボットを用いた専門検索エンジンであり，これらのシステムではローカルデータベースを構築し，様々な機械学習や知識表現アルゴリズムをデータに適用することにより優れた検索機能を提供している．しかしながら個人のホームページやレシピのような多くのサイト上に広く分散している Web ページに対しては，ページの収集にかかる時間やネットワーク帯域を消費する点から個々の専門検索エンジンがロボットを用いるのは困難であり，この手法は Web サイトの数が限られたドメインだけに適した方法といえる．

専門検索エンジンの構築に汎用検索エンジンの巨大なインデックスを利用することは有効な手法である[5]．たとえば Ahoy!☆☆[11] は個人ホームページの検索に特化した検索エンジンであり，このシステムはユーザの

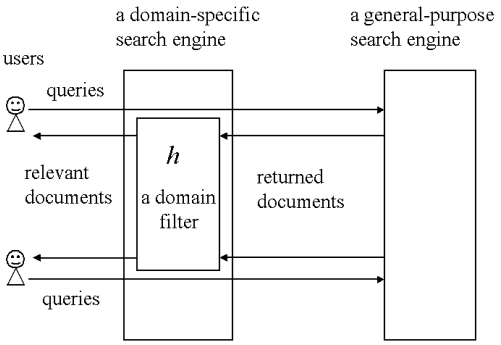

図 1 フィルタリングモデルを用いた専門検索エンジン構築モデル
Fig. 1 The filtering model of building domain-specific Web search engines.

クエリを汎用検索エンジンに転送する．そして得られた結果からドメインに特化したフィルタを用いて不必要なページを削除する．本論文ではこのような専門検索エンジン構築手法を “フィルタリングモデル” と呼ぶ（**図 1**）．さらに Ahoy! は，以前の成功した検索からドメインに属する Web ページの URL のパターンを学習する機能を有している．しかし全体としての正確さは基本的に人間の作り込んだ知識に依存しており，他のドメインへの適用も困難である．

いくつかのサンプルからドメインフィルタを自動的に構築して，ドキュメントを分類する自動テキストフィルタリングは情報検索[1] や機械学習[7] の分野における主要な研究テーマの 1 つである．ただしこれらのテキスト分類の研究のほとんどは，電子メールやネットニュースのように求めたいドメインに属するテキストの比率が高い集合に限定して適用されている．これに対して WWW の場合，そこからランダムにサンプリングしたとしてもドメインに属するページ（正例）が含まれる可能性はほとんどないという，訓練集合作成の問題が存在する．

このように従来のテキスト分類の手法を WWW の専門検索エンジン構築に単純に適用することはできない．そこで我々はユーザの入力すると予想されるキーワードを含む Web ページだけを対象として考えることで，WWW からドメインに関連する Web ページを多く収集する方式を採用した．

本論文では，まず 2 章で検索隠し味を用いた専門検索エンジン構築手法について提案した後，3 章で検索隠し味を抽出する機械学習アルゴリズムについて述べる．そして 4 章でその手法の適用例を示し，最後に，5 章でその例を用いた評価実験を行う．

☆ http://cora.whizbang.com/
☆☆ http://ahoy.cs.washington.edu:6060/

2. 検索隠し味を用いた専門検索エンジン構築モデル

ここで専門検索エンジンの構築を機械学習の問題として定義する．$\mathcal{D}$ をすべての Web ページ，$\mathcal{D}_t$ を求めたいドメインの Web ページとすると，ある Web ページ $d \in \mathcal{D}$ を完全に分類する理想的なドメインフィルタは次のように定義される．

$$f(d) = \begin{cases} 1 & \text{if } d \in \mathcal{D}_t \\ 0 & \text{otherwise} \end{cases}$$

さらに $\mathcal{K}$ をドメイン中のすべてのキーワード集合とし，あるキーワード $k \in \mathcal{K}$ をブール変数としたときのすべてのブール式からなる仮説空間を $\mathcal{H}$ と定義する．ここで我々がブール仮説空間を考えた理由は，多くの汎用検索エンジンがブール式のクエリをサポートしているからである．

キーワードが Web ページに含まれるときにそのキーワード（ブール変数）に 1 を，それ以外のときに 0 を割り当てることで，キーワードのブール式は $\mathcal{D}$ を $\{0,1\}$ に写像する関数と見なすことができる．そしてフィルタリングモデルにおいてドメインフィルタ構築の問題は，誤り率

$$\frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} \delta(h(d), f(d))$$

$$\delta(h(d), f(d)) = \begin{cases} 1 & \text{if } h(d) \neq f(d) \\ 0 & \text{otherwise} \end{cases}$$

を最小にする仮説 h を見つけることと等価と考えられる．

本論文で提案する手法では，検索エンジンであればユーザは最初に何らかのキーワードを入力するという点に着目し，すべての Web ページではなくユーザの入力するキーワードを含む Web ページだけをドメインフィルタ構築のための学習の対象として考える．つまりサンプリングすべき範囲をすべての Web ページ集合 $\mathcal{D}$ からキーワード k を含む Web ページ集合 $\mathcal{D}(k)$ に減らす（**図 2**）ことで，サンプリング集合中のドメインに関係する Web ページ $\{d|(k \wedge h)(d) = 1\}$ の比率を高くする．それによりランダムサンプリングでは不可能な訓練集合作成の問題を解決し，学習アルゴリズムを適用することが可能となる．

図 1 のように，ここで学習されたドメインフィルタ h を用いて汎用検索エンジンの検索結果をフィルタリングすれば，ドメインに属するページを得ることができる．しかし，実は**図 3** のようにユーザの入力したクエリ k とドメインフィルタ h の連言をとり，クエリを $k \wedge h$ に修正して汎用検索エンジンに投入すること

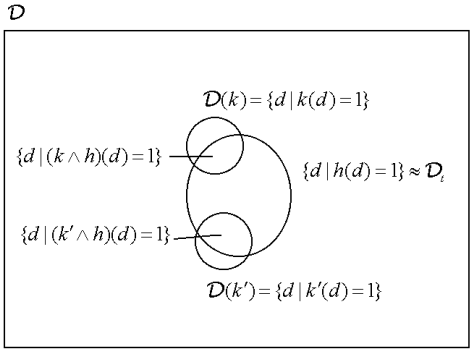

図 2 キーワードを含む Web ページのサンプリング
Fig. 2 Sampling with input keywords to increase the ratio of positive examples.

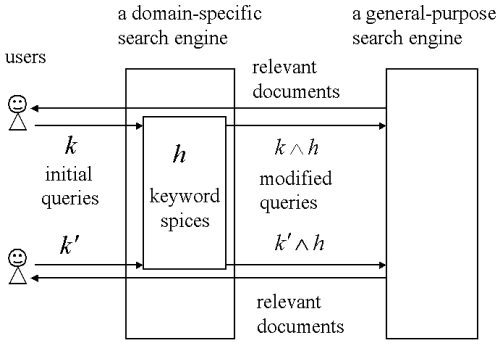

図 3 検索隠し味を用いた専門検索エンジン構築モデル
Fig. 3 The keyword spice model of building domain-specific Web search engines.

で，k を含みかつドメインに属するページだけを直接得ることができる．これは上のフィルタリングモデルとはまったく反対のモデルであり，h をフィルタではなく “検索隠し味” として用いることで，フィルタリングモデルのように，無関係な Web ページまでいったん専門検索エンジンの側に取得してからフィルタにかけるという余分な処理が必要でなくなるのである．

3. 検索隠し味抽出アルゴリズム

3.1 サンプルページの収集

ある特定のキーワード k（たとえば牛肉など）の検索隠し味を見つけることは比較的容易である．なぜならそのキーワードを汎用検索エンジンに入力して得られる結果だけから，学習を行えばよいからである．しかしあるドメインのための検索隠し味は，将来ユーザが入力すると予想されるキーワードすべてに対して十分な効果を持つ必要がある．

$p(k)$ をユーザがその専門検索エンジンに入力するキーワード k の確率分布とすると，そのシステムの誤り率の期待値は次の式で表される．

$$\sum_{k\in\mathcal{K}} p(k) \sum_{d\in\mathcal{D}(k)} \frac{1}{|\mathcal{D}(k)|}\delta((k\wedge h)(d), f(d))$$

そしてこの値を最小化するブール式が最も効果的な検索隠し味といえ，それを求めるためには学習の訓練集合に $p(k)$ を用いる必要がある．しかし事前に $p(k)$ を得ることは難しいため，適当な $p(k)$ から始めて，ユーザが入力したキーワードを収集することで $p(k)$ の修正を行っていくことが妥当といえる．

本論文では適用例としてレシピ専門検索エンジンの構築を試みた．その際，料理ドメインの食材リストからいくつかのキーワードを適当に選択したが，それぞれのキーワードの生起確率は同等と見なし，どのキーワードについても同数のサンプル Web ページを収集した．

3.2 決定木学習を用いた検索隠し味の導出

まず最初に後の手順で利用するために，収集した Web ページを訓練集合 $\mathcal{D}_{training}$ と検証集合 $\mathcal{D}_{validation}$ の 2 つの集合に分割する．このとき Web ページがドメインに属するかどうかは問題としない．

そして訓練集合に対して，ID3[8] で使用されている情報量に基づく決定木学習アルゴリズムを適用して決定木を得る．ただしここでは決定木の枝刈りは行わない．学習手法として決定木を使用した理由は，決定木は多くの検索エンジンがサポートするブール式に容易に変換することができるからである．学習における属性（キーワード）の数が十分に多いため，ここで得られる決定木は訓練集合中のすべての Web ページがドメインに属するかどうかを正確に分類する．

図 4 に単純な決定木の例を示す．決定木において各キーワードは 1（Web ページがそのキーワードを含むとき）と 0（含まないとき）の値を持つ属性として扱われる．そして図 4 の決定木は Web ページを T（料理レシピページ）と F（料理レシピ以外のページ）の 2 つのクラスに分類するものであり，たとえば “大さじ” を含まず “作り方” を含み “家庭” を含まず “トップ” を含まない Web ページはクラス T に分類される．

次に検索エンジンに入力できるブール式を作るために決定木のノードから葉へのパスをルール化する．ただしドメインに属する Web ページが抜き出せればよいので，ドメインに属するというクラス T の葉へのパスだけを選択する．ここで各ルールは，パス上の各ノードでのキーワードを含むか含まないかの条件判定（リテラル）の連言で条件部が表現される連言ルールとなる．そしてそれらのルールの条件部の選言をとったブール式の選言標準形が訓練集合のドメインに属するページだけをすべて抜き出す検索式，すなわち “検索隠し味” となる．図 4 の決定木では

> 大さじ OR（NOT 大さじ AND 作り方 AND NOT 家庭 AND NOTッップ）OR（NOT 大さじ AND NOT 作り方 AND こしょう AND NOT 鍋）

がその式となる．しかし一般的に決定木は過学習のため大きくなり，そこから導かれるブール式は複雑で汎用検索エンジンに入力できない．そのため決定木やルールの単純化が必要となる．

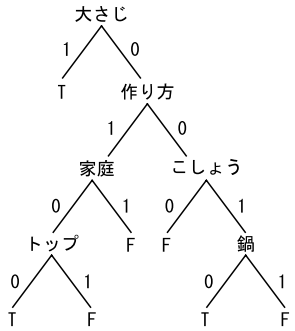

図 4 決定木の例
Fig. 4 A simple decision tree.

3.3 検索隠し味の単純化

決定木の枝刈り（単純化）には統計を用いた手法，Reduced-error pruning[9] などがあるが，本アルゴリズムでは次に示す利点[7] により決定木をルール化してから単純化する Rule post-pruning[10] を元にしたアルゴリズムを採用している．

- ルールが異なれば条件判定をまったく別のものとして対応できる（木では 1 つの条件判定（ノード）が複数のルールに影響を及ぼしている）．
- 1 つのルール内の条件判定はすべて同じように削除できる（木ではルートに近いノードほど削除するのが難しい）．
- 人間が理解しやすい．

ただし条件判定の削除の指標には一般的に用いられる誤り率ではなく，次に示す調和平均（Harmonic mean）[12] の値を用いた．

$\mathcal{D}_{recipe}$ と $\mathcal{D}_{rule}$ をそれぞれ，検証集合 $\mathcal{D}_{validation}$ 中の Web ページを人間がドメインに属すると判断したページ集合と，ルールが判断したページ集合とすると，検証集合に対するそのルールの適合率 $P_{validation}$ と再現率 $R_{validation}$ は次のように定義される．

$$P_{validation} = \frac{|\mathcal{D}_{rule}\cap\mathcal{D}_{recipe}|}{|\mathcal{D}_{rule}|}$$

$$R_{validation} = \frac{|\mathcal{D}_{rule}\cap\mathcal{D}_{recipe}|}{|\mathcal{D}_{recipe}|}$$

そして $\mathcal{D}_{validation}$ における調和平均 F_r は次の式で示される．

$$F_r = \frac{2}{\frac{1}{R_{validation}} + \frac{1}{P_{validation}}}$$

3.3.1 連言項からのリテラルの除去による単純化

本アルゴリズムではまず，決定木から得られた各ルールに対して，この F_r の値が小さくならないように，ルールの条件部である連言項からキーワードを含むか含まないかの条件（リテラル）を取り除くことで単純化を行う．

ここで一般に Rule post-pruning の指標として用いられる誤り率 E の代わりに調和平均 F_r を用いることの有効性について考える．誤り率 E を用いた場合，E が減少するときにリテラルを削除することができ，前述の記号を用いると E は

$$E = \frac{\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|}{|\mathcal{D}_{rule}|}$$

と表される．連言項からリテラルが削除されると $|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|$ および $|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}|$ の値は必ず増加する．そこでそれぞれの増加量を $\delta|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|$ および $\delta|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}|$ とすると，誤り率 E が減少する条件は

$$\frac{\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|}{|\mathcal{D}_{rule}|} \geq \frac{\delta\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|}{\delta\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right| + \delta|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|}$$

となり，これは次の条件式に変換できる．

$$\delta\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right| \leq \frac{\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|}{|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|} \times \delta|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|$$

単純化前の連言項においては一般的にこの条件式の右辺の第 1 項の値は非常に小さいため，E を減少させるには $\delta\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|$ が非常に小さく $\delta|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|$ が大きなリテラルを削除する必要がある．しかしこのようなリテラルは連言項にはあまり存在しない．さらに誤り率を用いた場合，連言項の最初の誤り率が 0 であると，その連言項を満たすドメインに属するページがどれだけ少なくてもリテラルを削除することができない．

次に調和平均 F_r を用いた場合について考える．$|\mathcal{D}_{rule}| = |\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}| + \left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|$ であるため，F_r の分母 f_r は次のように表される．

$$\begin{aligned} f_r &= \frac{1}{R_{validation}} + \frac{1}{P_{validation}} \\ &= \frac{|\mathcal{D}_{rule}| + |\mathcal{D}_{recipe}|}{|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|} \\ &= 1 + \frac{\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|}{|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|} + \frac{|\mathcal{D}_{recipe}|}{|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|} \end{aligned}$$

そして f_r が小さくなる（F_r が大きくなる）場合，そのリテラルを削除することができる．

ここで f_r の第 2 項が減少する条件は，

$$\frac{\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|}{|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|} \geq \frac{\delta\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|}{\delta|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|}$$

であり，次の条件式に変形できる．

$$\delta\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right| \leq \frac{\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|}{|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|} \times \delta|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|$$

これは前述した E と同じ条件であり，f_r の第 2 項を減らすリテラルはほとんどないことを示している．しかし f_r では第 3 項が分母だけが増加するため必ず減少し，第 2 項が増加した場合でも，その値が第 3 項の減少以下であればそのリテラルを削除できる．その結果，調和平均を指標に用いると，ドメインに属さないページの増加（$\delta\left|\mathcal{D}_{rule} \cap \overline{\mathcal{D}_{recipe}}\right|$）をある程度許しながら，できるだけドメインに属するページの増加（$\delta|\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}|$）を多くするようにリテラルを取り除くことができる．

ただし調和平均を用いる場合の問題点として，最初の連言項を満たすドメインに属するページ数が非常に少ない，すなわち $R_{validation}$ が小さい場合，この値を大きくすることで F_r を大きくする作用のために，多くのドメインに属するページが満たす反面，ドメインに属さないページも多くが満たす連言項が生成される．たとえば後述する決定木から導かれる連言項 “NOT 材料 AND NOT 大さじ AND NOT 沸騰 AND 適量” は，1 ページのドメインに属するページだけしか満たさない．しかし調和平均を用いてリテラルの削除を行うと最終的に “NOT 沸騰” という 263 ページのドメインに属するページと 704 ページの属さないページが満たす連言項が生成されてしまう．ただしこのような連言項は次に示す選言標準形の単純化において削除されるため，最終的には問題とならない．

3.3.2 選言標準形からの連言項の除去による単純化

次の手順として，単純化された複数の連言項の選言をとることで，1 つのブール式の選言標準形 h を生成する．これが検索隠し味の初期状態であるが，まだ検索エンジンに入力するには大きく，上述の不適切な連言項も含んでいる．そこでさらにこの選言標準形 h に対して，検証集合 $\mathcal{D}_{validation}$ とその調和平均 F_s を

用いた単純化を行う．

ここで $\mathcal{D}_{spice}$ を選言標準形 h によってドメインに属すると分類された Web ページとすると $\mathcal{D}_{validation}$ に対する h の適合率 $P_{validation}$ および再現率 $R_{validation}$ は次のように定義される．

$$P_{validation} = \frac{|\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}|}{|\mathcal{D}_{spice}|}$$

$$R_{validation} = \frac{|\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}|}{|\mathcal{D}_{recipe}|}$$

そしてこれらの調和平均 F_s が大きくなる場合に，選言標準形から連言項を取り除き，どの連言項も削除できなくなった h が検索隠し味となる．

ここでも選言標準形から連言項を取り除く指標として調和平均を用いることの有効性について考える．F_s の分母 f_s は次のようになり，

$$f_s = 1 + \frac{\left|\mathcal{D}_{spice} \cap \overline{\mathcal{D}_{recipe}}\right| + |\mathcal{D}_{recipe}|}{|\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}|}$$

この値が減少する（調和平均が増大する）ように連言項を取り除く．

選言標準形から連言項を取り除くため $|\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}|$ と $|\mathcal{D}_{spice} \cap \overline{\mathcal{D}_{recipe}}|$ は必ず減少する．そこでそれぞれの減少量を $\delta|\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}|$ および $\delta|\mathcal{D}_{spice} \cap \overline{\mathcal{D}_{recipe}}|$ とすると，連言項を削除できる条件は次のようになる．

$$\delta\left|\mathcal{D}_{spice} \cap \overline{\mathcal{D}_{recipe}}\right| \geq \frac{\left|\mathcal{D}_{spice} \cap \overline{\mathcal{D}_{recipe}}\right| + |\mathcal{D}_{recipe}|}{|\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}|} \times \delta|\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}|$$

この条件が示しているのは，ドメインに属するページの減少（$\delta|\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}|$）はわずかで，ドメインに属さないページの減少（$\delta\left|\mathcal{D}_{spice} \cap \overline{\mathcal{D}_{recipe}}\right|$）を大きくする連言項が削除されるということである．先に述べたようにいくつかの連言項は多くのドメインに属するページが満たす反面，ドメインに属さないページも多くが満たしてしまっている．しかしドメインに属するページは他の連言項も満たしている場合が多いため，問題のある連言項はこの条件により削除される．そして最終的に単純な検索隠し味が生成される．

最終的な検索隠し味抽出アルゴリズムを**図 5** に示す．

4. 料理レシピドメインへの適用

前章で述べたように，我々は本手法の適用例として料理レシピドメインを選択し，いくつかのキーワードを最終的に利用する汎用検索エンジン “goo” に投入することで Web ページの収集を行った（**表 1**）．ただし収集した Web ページがレシピページかどうかは人間が Web ページを見ることで判断した．

そしてこれらの収集した Web ページを，訓練集合と検証集合に分割した．この際 Web ページがレシピページかどうか，またどのキーワードを用いて収集されたページかということは無視して，ランダムに分割を行った．

その訓練集合に対して決定木学習アルゴリズムを用いてできた決定木を**図 6** に示す．この決定木はノード数 45 であり，ここから単純に導かれるブール式は

(0) ユーザのクエリの予想分布 $p(k)$ から入力キーワード k を決め，そのキーワードを含む Web ページを収集する．そしてそれぞれがドメインに属するページかそうでないかをチェックする．

(1) Web ページ集合を初期決定木を作るための訓練集合 $\mathcal{D}_{training}$ とブール式の単純化を行うための検証集合 $\mathcal{D}_{validation}$ に分割する．

(2) $\mathcal{D}_{training}$ から情報量に基づく決定木学習アルゴリズムを用いて初期決定木を作る．

(3) 決定木において，ドメインに属する Web ページを分類するノードから葉へのパスをそれぞれ連言ルールに変換する．

(4) **For each** ルール r **do**

Repeat

次の調和平均 F_r の値が最も増加するようにルールの条件部からリテラルを取り除く．

$$F_r = \frac{2}{\frac{1}{R_{validation}} + \frac{1}{P_{validation}}}$$

ここで

$P_{validation} = |\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}| / |\mathcal{D}_{rule}|$

$R_{validation} = |\mathcal{D}_{rule} \cap \mathcal{D}_{recipe}| / |\mathcal{D}_{recipe}|$

であり，$\mathcal{D}_{recipe}$ と $\mathcal{D}_{rule}$ はそれぞれ，検証集合 $\mathcal{D}_{validation}$ 中で，人間がドメインに属すると分類したページ集合，およびルール r がドメインに属すると分類したページ集合である．

Until どのリテラルを削除しても F_r が減少する．

End

(5) すべてのルールの条件部の選言をとることでブール式の選言標準形 h を得る．

(6) **Repeat** 次の調和平均 F_s の値が最も増加するように選言標準形 h から連言項（もとのルールの条件部）を取り除く．

$$F_s = \frac{2}{\frac{1}{R_{validation}} + \frac{1}{P_{validation}}}$$

このとき

$P_{validation} = |\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}| / |\mathcal{D}_{spice}|$

$R_{validation} = |\mathcal{D}_{spice} \cap \mathcal{D}_{recipe}| / |\mathcal{D}_{recipe}|$

ここで $\mathcal{D}_{spice}$ は検証集合 $\mathcal{D}_{validation}$ 中で h がドメインに属すると分類したページ集合である．

Until どの連言項を削除しても F_s が減少する．

Return h

図 5 検索隠し味抽出アルゴリズム
Fig. 5 The keyword spice extraction algorithm.

表 1 レシピドメインのための収集 Web ページ
Table 1 Collected Web documents in the cooking domain.

キーワード	レシピ	レシピ以外	合計
牛肉	47	153	200
鶏肉	88	112	200
ピーマン	79	121	200
じゃがいも	49	151	200
かぼちゃ	42	158	200
大根	64	136	200
鮭	15	185	200
豆腐	45	155	200
トマト	33	167	200
白身魚	103	97	200
合計	565	1435	2000

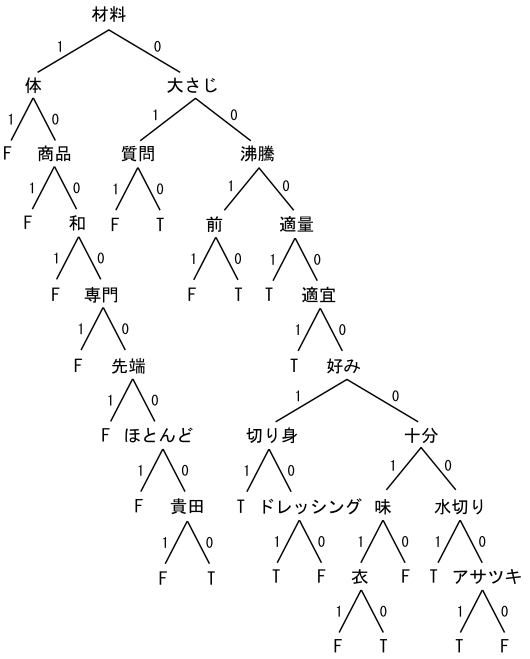

図 6 訓練集合から得られた決定木
Fig. 6 A decision tree of the training set.

ルール数が 10 でキーワード数が 65 の非常に複雑なものとなる．しかしこの決定木をルールに置き換え，検証集合に対する調和平均を指標として単純化する前章のアルゴリズムを適用した結果，次の検索隠し味が抽出された．

- （材料 AND NOT 専門 AND NOT 商品）OR（大さじ）

これはキーワード数 4 と大きく単純化されており，ブール式をサポートした検索エンジンのどれにでも入力することができる．

今回我々は収集 Web ページの訓練集合と検証集合の分割を上記以外にも 4 回行い，同様に検索隠し味を求めた．各分割に対してアルゴリズムを適用した際の

表 2 単純化の結果
Table 2 Pruning result.

		分割				
		1	2	3	4	5
決定木	ノード数	45	55	47	49	49
	キーワード数	65	89	76	87	62
手順（4）	ルール数	10	15	13	15	10
	キーワード数	17	32	26	34	19
手順（6）	連言項の数	2	2	2	2	2
	キーワード数	4	3	4	4	4

表 3 抽出された検索隠し味
Table 3 Extracted keyword spices.

分割	隠し味
1	（材料 AND NOT 専門 AND NOT 商品）OR（大さじ）
2	（材料 AND NOT 東京）OR（大さじ）
3	（材料 AND NOT 商品 AND NOT 結果）OR（大さじ）
4	（材料 AND NOT 発生 AND NOT 商品）OR（調味）
5	（材料 AND NOT 季節 AND NOT 説明）OR（大さじ）

単純化の結果を**表 2** に示す．また，各分割で抽出された検索隠し味をまとめて**表 3** に示す．それぞれ異なる訓練・検証集合の組から導かれたものであるが，ほぼ同じキーワード数であり，さらに含まれるキーワードも同じものが多かった．

5. 評価実験

前章で求めた検索隠し味（1 番目）を使って，料理レシピドメインにおいて検索隠し味を用いた手法の検索能力の評価実験を行った．このとき汎用検索エンジンには goo を使用し，ユーザが入力するキーワードとして検索隠し味を抽出する Web ページ収集に使用しなかった “豚肉” “ほうれん草” “エビ” の 3 つのキーワードを用いた．

5.1 適合率

3 つのキーワードについて，goo にキーワードだけを入力した場合と検索隠し味を付加して入力した場合の結果トップ 1,000 件を見たときの適合率，すなわちレシピページの割合を調べた．**図 7** にキーワードが “豚肉” の場合の，検索結果の上位から累積した Web ページにおける適合率の推移を示す．また最終的な結果を**表 4** に示す．

表 4 から検索隠し味によって適合率が飛躍的に向上したことが分かる．

5.2 推定再現率

次に我々はこの手法により再現率がどう変化するかの評価も行った．再現率とは検索されなければいけない Web ページのうち，実際に検索できたものの割合であり，これを無視すれば適合率を高くすることは容易である．ただし WWW 中の全レシピページの数 $\mathcal{D}_t$

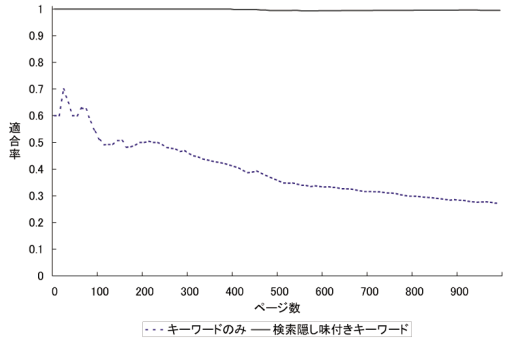

図 7 キーワード “豚肉” に対する goo の検索結果上位ページにおける適合率
Fig. 7 Precision over the top ranked pages returned by goo for the keyword “pork”.

表 4 goo の検索結果上位 1,000 ページの適合率
Table 4 Precision over the top 1,000 pages returned by goo.

キーワード	キーワードのみ	検索隠し味付きキーワード
豚肉	0.271	0.995
ほうれん草	0.205	0.979
エビ	0.063	0.986

が分からないため，正確な再現率を求めるのは困難である．そこで我々は汎用検索エンジンの結果から再現率を推定する方法を考案した．まず検索結果として表示されるクエリに適合する Web ページの総数（ヒット数）と，先に求めた 1,000 件の適合率から，入力キーワードを含むレシピページの総数 $Reldoc_{index}$ を次の式で推定する．

$Reldoc_{index} \simeq$
（ヒット数）×（1,000 件の適合率）

同様にして検索隠し味を付加した場合に検索されたレシピページの総数は

$Reldoc_{spice} \simeq$
（検索隠し味を付加したときのヒット数）
×（検索隠し味を付加したときの 1,000 件の適合率）

と表される．

ここですべての Web ページを知ることは不可能であるので，検索エンジンで検索できる Web ページを全 Web ページと見なすこととする．つまり $Reldoc_{index}$ があるキーワードを含みドメインに属する（ここならレシピ）ページのすべてということとする．そのとき検索隠し味を付加した場合の再現率は次の式で表される．

$$R \simeq \frac{Reldoc_{spice}}{Reldoc_{index}}$$

表 5 goo の結果を利用した推定再現率
Table 5 Estimated recall of the queries with keyword spices over the index of goo.

Query	$Reldoc_{index}$	$Reldoc_{spice}$	推定再現率
豚肉	10728	10084	0.9400
ほうれん草	4744	4126	0.8695
エビ	5868	5728	0.9761

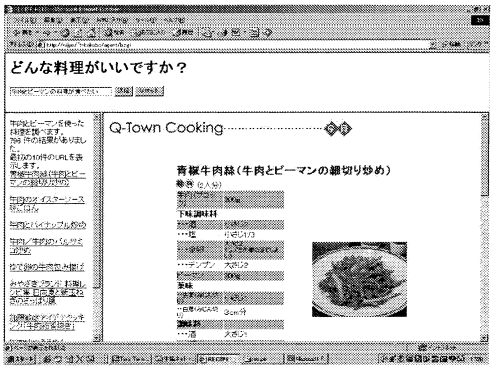

図 8 料理レシピ専門検索エンジン
Fig. 8 Recipe search engine.

表 5 は goo の検索結果と先の適合率を利用して求めた推定再現率である．この表から再現率に関しても 86%以上と，高い値を保っていることが分かる．つまり我々の用いた手法がドメインに属さない Web ページだけを排除して，ドメインに属するページはほとんど取り除いていないことが分かる．

以上の結果から検索隠し味を用いた手法が，専門検索エンジンを構築する手法として非常に効果的であることが確かめられた．

ここで 1 章で述べた “塩” を検索隠し味として，“豚肉” のキーワードに対して同様に適合率と推定再現率を調べたところ，それぞれ 0.674 と 0.871 であった．これは我々のアルゴリズムを用いることで，より性能の良い検索隠し味を得られたことを示している．

図 8 に検索隠し味を使った料理レシピ専門検索エンジンの画面を示す．これは自然言語でのユーザ入力に対して，形態素解析により食材名だけを抜き出した後，上述の適用例で得た検索隠し味を追加して goo に転送することで，その食材を使った料理レシピを検索して表示するシステムである．

6. おわりに

本論文では新たな専門検索エンジン構築手法として，検索隠し味というブール式をユーザの入力クエリに加えることで，汎用検索エンジンの検索結果を向上させる方法について提案した．そして本手法により従来の

システムで必要とされていた人間による知識の作りこみなしに容易に専門検索エンジンを構築することができる．また検索隠し味を Web ページ集合から抽出する，学習アルゴリズムについて述べた．このアルゴリズムにより一般の検索エンジンでサポートできる単純なブール式を検索隠し味として抜き出すことが可能となる．そして実際にレシピドメインで検索隠し味を抽出し，それを用いた評価実験から，本手法により適合率を飛躍的に向上させることが確認できた．さらに検索エンジンの結果を用いて再現率の推定を行い，再現率の低下もわずかであることを確かめた．

今回は本手法の適用例としてレシピドメインを選択したが，今後はレストランや個人ホームページといった他のドメインについても専門検索エンジンを構築する予定である．さらに現在は手動で行っている訓練集合の Web ページのドメインに対するクラス分けに関しても，ディレクトリ型検索エンジンなどに含まれる情報を利用することで，全手順を自動のアルゴリズムにすることを検討している．

謝辞 本研究はイメージ情報科学研究所，および文部省科学研究費基盤研究（A）「コミュニティ情報流通プラットフォームの構築」（平成 11 年度〜13 年度，課題番号 11358004）から援助を受けて行われました．

参 考 文 献

1) Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, Addison-Wesley (1999).
2) Butler, D.: Never trust a human, *Nature*, Vol.405, p.115 (2000).
3) Cohen, W.W.: A Web-based Information System that Reasons with Structured Collections of Text, *Agents'98*, pp.116–123 (1998).
4) Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K. and Slattery, S.: Learning to Extract Symbolic Knowledge from the World Wide Web, *AAAI-98*, pp.509–516 (1998).
5) Etzioni, O.: Moving Up the Information Food Chain: Deploying Softbots on the World Wide Web, *AAAI-96*, pp.1322–1326 (1996).
6) McCallum, A., Nigam, K., Rennie, J. and Seymore, K.: A Machine Learning Approach to Building Domain-Specific Search Engines, *IJCAI-99*, pp.662–667 (1999).
7) Mitchell, T.M.: *Machine Learning*, McGraw-Hill (1997).
8) Quinlan, J.R.: Induction of Decision Trees, *Machine Learning*, Vol.1, pp.81–106 (1986).
9) Quinlan, J.R.: Simplifying decision trees, *International Journal of Man-Machine Studies*, Vol.27, pp.221–234 (1987).
10) Quinlan, J.R.: *C4.5 : Programs for Machine Learning*, Morgan Kaufmann Publishers, Inc. (1993).
11) Shakes, J., Langheinrich, M. and Etzioni, O.: Dynamic Reference Sifting: A Case Study in the Homepage Domain, *6th International World Wide Web Conference*, Santa Clara and CA (1997).
12) Shaw Jr., W.M., Burgin, R. and Howell, P.: Performance Standards and Evaluations in IR Test Collections: Cluster-Based Retrieval Models, *Information Processing & Management*, Vol.33, No.1, pp.1–14 (1997).

(平成 13 年 5 月 10 日受付)
(平成 14 年 3 月 14 日採録)

小久保 卓

平成 11 年京都大学工学部情報学科卒業．平成 13 年京都大学大学院情報学研究科社会情報学専攻修士課程修了．同年 NTT ドコモ（株）入社．

小山 聡

平成 6 年京都大学工学部数理工学科卒業．平成 8 年京都大学大学院工学研究科数理工学専攻修士課程修了．平成 14 年京都大学大学院情報学研究科社会情報学専攻博士後期課程修了．博士（情報学）．平成 8 年〜10 年日本電信電話株式会社勤務．平成 13 年〜14 年日本学術振興会特別研究員．現在，京都大学大学院情報学研究科社会情報学専攻助手．情報検索，人工知能の研究に従事．電子情報通信学会，人工知能学会，ACM，AAAI 各会員．

山田 晃弘

昭和 60 年岡山大学工学部電子工学科卒業．同年三洋電機（株）入社．現在，三洋電機（株）研究開発本部主任企画員として情報検索の研究開発に従事．電子情報通信学会会員．

北村 泰彦（正会員）

昭和 58 年大阪大学基礎工学部情報工学科卒業．昭和 63 年大阪大学大学院博士課程修了．工学博士．同年大阪市立大学工学部電気工学科助手．現在，大阪市立大学大学院工学研究科情報工学専攻助教授．マルチエージェントシステム，ヒューリスティック探索，WWW 情報統合の研究に従事．IEEE，AAAI，ACM，人工知能学会，電子情報通信学会，ソフトウェア科学会等の会員．

石田 亨（正会員）

昭和 51 年京都大学工学部情報工学科卒業．昭和 53 京都大学大学院修士課程修了．同年日本電信電話公社電気通信研究所入所．現在，京都大学大学院情報学研究科社会情報学専攻教授．工学博士．IEEE Fellow．人工知能，コミュニケーション，社会情報システムに興味を持つ．