情報源の選択に基づくウェブからのレコード抽出手法

張　建偉† 黒川　沙弓† 石川　佳治†,†† 北川　博之†,††

† 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻
†† 筑波大学計算科学研究センター
〒 305–0044 茨城県つくば市天王台 1–1–1
E-mail: {zjw,saku39}@kde.cs.tsukuba.ac.jp,{ishikawa,kitagawa}@cs.tsukuba.ac.jp

あらまし　現在，膨大な情報がネットワーク上に存在しており，ユーザが関心を持つ情報をそこからいかに効率的に抽出するかが重要となっている．本稿では，ユーザが提示する例示レコード集合をもとに，内容の関連が深いレコードを選択的に抽出するアプローチを提案する．効率的なレコードの抽出のためには，関連するレコードを含んでいる可能性が高い文書群を特定することが重要となる．そのため本手法では，例示レコード集合をもとに文書リポジトリ中の文書からサンプルを抽出し，サンプルからのレコード抽出の結果をもとに抽出対象の文書群を選択する．また，質の高いレコード抽出のためには，抽出結果におけるノイズの削減が重要となる．このため，本研究では，レコード抽出とデータクリーニングの処理を統合し，あいまいさを持つ抽出結果に対しユーザからのフィードバックを受ける方式を提案する．これらの機能により，大量の文書に対しても，ユーザの要求に合ったレコードを効率的に抽出することを目指している．

キーワード　情報抽出，情報源選択，データクリーニング，フィードバック

Extracting Records from the Web Based on Web Resource Selection

Jianwei ZHANG†, Sayumi KUROKAWA†, Yoshiharu ISHIKAWA†,††, and
Hiroyuki KITAGAWA†,††

† Department of Computer Science, Graduate School of Systems and Information Engineering
†† Center for Computational Sciences
University of Tsukuba
1–1–1 Tennoudai, Tsukuba, Ibaraki, 305–8573 Japan
E-mail: {zjw,saku39}@kde.cs.tsukuba.ac.jp,{ishikawa,kitagawa}@cs.tsukuba.ac.jp

Abstract　There is a vast amount of valuable information on the network. It is important to efficiently extract information in which the user is interested from the Web. In this paper, we propose an approach that selectively extracts records whose contents are deeply relative to the example record set given by the user. For an efficient extraction, it is necessary to locate the documents with high possibility of containing the relative records. Therefore our approach samples the document repository and runs the record extraction based on this sample. Then the document set is selected based on the result of extraction. Moreover, the elimination of noise in the extraction result is also important for a high-quality record extraction. Our approach integrates the record extraction and data cleaning, and receives feedback for the vague records from the user. Using these techniques we aim to efficiently extract records suitable for user requirements even from a large amount of documents.

Key words　information extraction, resource selection, data cleaning, feedback

1. はじめに

今日では HTML 文書をはじめとする膨大な量の文書データがネットワーク上に存在しているが，それらの情報は構造化されていず，多種多様である．そのため，大量かつ異種性の高い文書データからの**情報抽出**（information extraction）の研究が近年重要視されている [1]〜[5]．これらの研究はウェブページやテキストデータにおける文書構造やテキストのパターンな

どを分析することで，質の高い情報抽出の実現を図っている．本稿では特に，構造化された**レコード**構造の情報を抽出しようとする試みに着目する．直感的には，レコードとは 1 ないし複数の属性からなるデータで，各属性には同種のデータが含まれるようなものをいう．抽出されたレコードの集合は一種のデータベースと考えられ，既存のデータベースとの統合との活用など，様々な形で応用することが可能となる．

一方では，近年**分散情報検索**（distributed information retrieval）に関する研究 [6], [7] が盛んに進められている．分散情報検索の重要なプロセスの一つとして，情報源の選択は検索の精度に大きく関わるものである．ユーザから与えられた問合せ，関連する情報を多く有する情報源に送られれば適切な情報が得られる可能性が高くなる．そのため，分散情報検索では適切な情報源を選択するという処理が特に重要となる．

本稿では，情報抽出に情報源選択の考えを導入し，ユーザが提示した例示レコード集合に関連する情報を含む情報源を選択的にアクセスするレコードの抽出手法を提案する．本研究では，情報抽出に DIPRE [1] を利用する．DIPRE は，与えられたサンプルレコードをもとに，文書集合（特に HTML 文書集合）から情報抽出のパターンを学習し，レコードを抽出する手法である．DIPRE は，情報抽出のために文書リポジトリ全体を繰り返しスキャンするため，大量のデータに対しては多大なコストを要する．本手法は，リポジトリからまずサンプル文書を選択し，そこからのレコード抽出の結果により，情報抽出の対象となるサイト集合の中でどのサイトが今後の抽出により有用かを予測する．抽出に有用と判断されたサイト内の文書を優先的に利用することで，より少ない文書数で多くのレコードの抽出を図る．DIPRE で抽出されたレコード集合には，ノイズを含んだレコードや，提示された例示レコード集合とあまり関連しないレコードが含まれていると考えられる．本研究では，前者に対しては，データクリーニングの処理を統合することにより，抽出されたレコード集合におけるノイズの削減を図る．後者の問題に対しては，抽出されたレコード集合のうち，曖昧性が高いレコードを中心にユーザからのフィードバックを受ける．これをもとに，レコード抽出パターンの修正およびサイトのスコアの修正を行う．

2. 関 連 研 究

2.1 情 報 抽 出

DIPRE (Dual Iterative Pattern Relation Expansion) [1] は文書集合からレコード集合を抽出するためのアプローチであり，HTML 文書からの情報抽出のために開発されたものである．例として，いくつかの本の著者とタイトルのペアがユーザにより与えられたとする．DIPRE は，文書集合の中から与えられたレコードを抽出するためのパターンの集合を抽出し，今度はそれらのパターンを用いて新たなレコードの抽出を図る．このようなプロセスを繰り返すことで，文書集合から多くのレコード集合を抽出しようと試みる．特にウェブ環境では，関連するレコードが HTML 文書に一定の文脈で繰り返して現れる傾向があるため，この手法は単純であるがうまく機能するといわれる．後述のように，本研究ではレコード抽出に DIPRE を利用する．

[5] は，DIPRE の繰り返し処理回数を減らすためのレコード抽出手法を提案している．この手法は 1 回のレコード抽出の実行（文書リポジトリ中の文書をすべてスキャンする）の後，抽出可能な全てのレコードの数を予測し，その内ですでに抽出されたレコードの割合（カバー率）を推定する．提案された手法では，レコードが抽出できなくなるまでではなく，カバー率が閾値に達成したら繰り返し処理を終了する．

他の手法の例を挙げる．**Snowball** [2] はプレーンテキスト文書からパターンを生成し，レコードを抽出する．DIPRE よりも厳密なパターンを採用し，また，レコードの評価基準も提案している．パターンとしては，文字列のみでなく，固有表現（組織名，地名など）も利用している．**MDR** [4] はユーザのサポートを必要としない手法であり，多数のデータレコードを含む 1 つのページから，自動的にレコードを抽出する．この手法は HTML 文書の木構造の分析に基づくアルゴリズムを利用している．**Wrapper Induction** [3] は，ウェブページに対してラッパーを自動的に構築するための手法である．ページと抽出したい属性が与えられたときに，区切りパターンを帰納学習により導出する．

2.2 情報源選択

近年，集中型の情報検索に対して，**分散情報検索**（distributed information retrieval）の研究が活発に行われている．分散情報検索では，検索要求に対し，検索処理を分散した各情報源で行い，各情報源から返ってきた結果を統合して検索結果とする．ユーザからのクエリを関連情報が含まれている情報源に優先して送るため，**情報源選択**（server selection）が必要となる．分散情報検索の研究においては，そのための手法が多く提案されている [8]〜[11].

メタサーチエンジン（metasearch engine）も分散情報検索に関連する技術の一種である [12], [13]．これは，複数のサーチエンジンに対して横断的な検索を行うための統一的なインターフェースを利用者に提供するシステムである．メタサーチエンジンの構成要素の一つであるデータベースセレクタ（database selector）は，利用者からの問合せに対して，有用と思われるサーチエンジンを選択する [14], [15].

本研究は，すべての文書集合を情報抽出の対象とするのではなく，サンプル文書に対する情報抽出の結果により，情報抽出用の情報源を選択する．情報源の選択の点はこれらの研究と関連が深い．

2.3 データクリーニング

データクリーニング（data cleaning or data cleansing）[16], [17] は，データの質を改良するためにデータから誤りや矛盾を取り除く処理であり，データマイニングの前処理の一環としてしられている．**Potter's Wheel** [18] は，対話的なデータクリーニングシステムである．ユーザはインタフェースを利用して，トランスフォームを行う．トランスフォームの効果は直ちにスクリーン上に示される．バックグラウンドでは，システムはユーザによって定義されたドメインに関してデータ

Author	Title
A.F. Cardenas	Data Base Management Systems
C.J. Date	An Introduction to Database Systems
D. Maier	The Theory of Relational Databases
H.S. Kunii	Graph Data Model and Its Data Language
R.G.G. Cattell	Object Data Management

図 1　例示レコード集合
Fig. 1　Example record set

値の構造を推論し，制限違反がないかどうかをチェックする．これにより，ユーザは複雑なプログラムを書く必要がなく，不整合が見つけられる時にトランスフォームを組み込むことで，データを修正する．

本研究では，異種性の高い文書データからのレコードの抽出を想定しているため，抽出されたレコードにノイズが含まれると考えられる．ノイズのあるレコードをもとに学習されたパターンは誤りが含まれる可能性が高い．そのため，抽出されたレコードを次の抽出処理に利用する以前に，データクリーニング処理を行うことが必要となる．データクリーニングにより，データ中のノイズを削減し，レコード抽出の精度を向上させる．

3. レコード抽出手法

本研究では，情報抽出には **DIPRE** (Dual Iterative Pattern Relation Expansion) [1] の利用を想定している．DIPRE の処理ステップは以下のようになる．

（1） シードとなるレコード集合が与えられる．たとえば，あるユーザがデータベース関係の本の著者とタイトルの情報に興味を持っている場合，図 1 に示すようなデータを提供することが考えられる．

（2） 文書リポジトリ（クローラで収集された HTML ページ集合あるいは他の種類の文書集合のことである）から，シード集合に対応するレコードの**オカレンス**（occurrence）の集合を見つける．オカレンスは

$$(url, attributes, order, prefix, separators, suffix)$$

という形式で表される．url はレコードが見つかった URL を表し，$attrbutes$ は抽出対象のレコードの配列を表す．たとえば図 1 の例の場合，$attributes[0]$ が著者名（例：C. J. Date）に，$attrbutes[1]$ がタイトル（例：An Introduction to Database Systems）に相当することになる．$order$ は，その文書内で属性がどの順序で出現したかを表す情報を含んでいる属性である．この例の場合は，対象の文書で，オカレンスが「著者，タイトル」という順で現れるか,「タイトル，著者」という順序で現れるかの区別を保持する．$prefix$ と $suffix$ はそれぞれ，最初および最後に出現する属性の前および後に出てくるタグ等のパターンである．$separators$ は属性間の区切りのパターンに相当し，図 1 の場合は $separators$ が著者とタイトル（あるいはタイトルと著者）を区切るパターンを保持することになる．

（3） 発見されたオカレンスの集合をもとにパターン集合を生成する．パターンは

$$(order, urlprefix, prefix, separators, suffix)$$

の形式を持つ．パターン生成においては，まず，オカレンス集合を同じ $order$ と $separators$ を持つオカレンスごとにグループ化する．含まれるオカレンスの数が 1 件しかないグループは削除し，残りの各グループについてパターン生成を試みる．

パターン生成においては，グループ内のすべてのオカレンスについて，url の最長接頭辞，$prefix$ の最長接尾辞，$suffix$ の最長接頭辞を抽出し，それぞれをパターンの $urlprefix, prefix, suffix$ とする．これらのいずれかが空になる場合，グループを分割して再度パターン抽出を試みる [1]．

（4） 追加されたパターン集合が得られると，これをもとに文書リポジトリを再度スキャンし，新たなレコードの抽出を試みる．そして，抽出されたレコードをシードレコード集合に追加する．このような処理を繰り返すことで，逐次的にレコードを抽出する．

図 2 に DIPRE の処理の流れを示す．

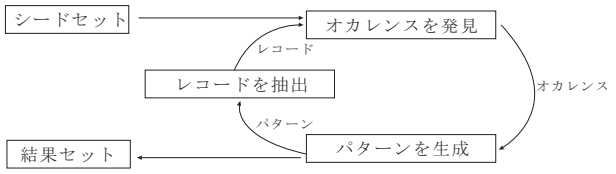

図 2　DIPRE 処理の流れ
Fig. 2　Flow of DIPRE

4. 情報源選択を融合した情報抽出手法

4.1 基本的な考え方

DIPRE では，静的な文書リポジトリを対象とし，収束するまでレコードの抽出を繰り返すということを基本としていた．オカレンスを発見するために文書リポジトリをスキャンする作業は必要であるが，文書リポジトリ中は一般的に膨大な量の文書があり，その処理には多大なコストがかかる．そこで本研究では，文書リポジトリ中の文書群から情報抽出により適した文書の部分集合を選択することを考える．文書リポジトリ中のすべての文書を利用したり，無選択的にリポジトリの一部分の文書を利用するのではなく，文書集合のサンプリングとそれに基づく有用度の予測により，有用な文書群を選び，抽出において優先的に利用する．これにより，少ない文書アクセス数で，例示データに関連するレコードを数多く抽出することを目的とする．

本研究では，レコード抽出のための文書リポジトリ中の文書が，**サイト**別にあらかじめ分けられているものとする．これは，ウェブから HTML 文書をクローリングしてきた場合，対象となったウェブサイト別にリポジトリに格納することに対応するが，必ずしも HTML 文書には限らない．たとえば，ニュースなどの文書データを提供する複数の情報源があった場合，トピックなどに違いがあると考えられるため，本研究ではそれらを独立した文書群として扱う．同じ情報源からの文書集合についても，たとえばニュース記事の場合は，記事に対応するカテゴリ

が付与される場合も多いことから，カテゴリ別に文書をグループ化することが考えられる．

図 3 には，本手法で想定するシステムの構成を示す．本手法では，以下のようなステップで情報抽出処理が行われる．

（1） まず，文書リポジトリから，初期の判断に用いられる文書集合の部分集合をランダムにサンプリングする．具体的には，文書リポジトリのすべてのサイトから，それぞれサイト内の指定された割合（例えば 10%）のページをランダムに取得する．選択された文書集合のことを**サンプル集合**と呼ぶ．

（2） サンプル集合に対して DIPRE の処理を適用する．次に DIPRE の結果として抽出されたレコード集合をもとに，各サイトのスコアを計算する．スコアの計算方法については後述する．

（3） DIPRE の処理により抽出されたレコードについて，トピックの点で不適合なもの，および，データの抽出にミスがあるものについてユーザからのフィードバックを受ける．これをもとに，レコード抽出パターンの修正，およびサイトのスコアの修正を図る．実際の処理においては，ユーザからのフィードバックの労力は多大なものとなる．そのため，本研究ではデータクリーニングの処理を統合し，ユーザによる判断を必要とするレコード数を削減する手法を検討している．

（4） その後，上位にランクされたサイトから順に，さらにページをサンプリングし，サンプル集合に追加する．更新されたサンプル集合において，DIPRE の処理を再開する．

このような処理を繰り返すことで，情報抽出の効率と精度の向上を図る．

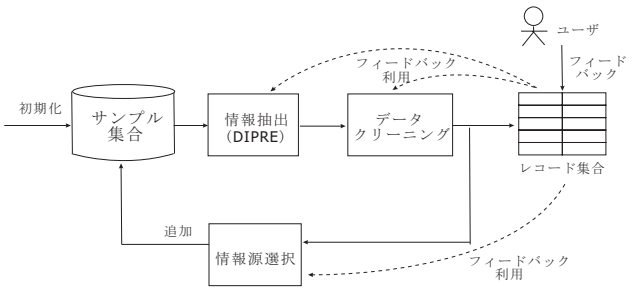

図 3　システムの構成

Fig. 3　System architecture

本研究においては，最初に選択されたサンプル文書集合に新たに文書が追加されることを想定している．そのため，サンプル集合が更新された後の次の DIPRE の処理では，新たな文書からレコードのオカレンスの抽出を試みることと，新しいパターンで既存のサンプル集合を探索しなおすことが必要となる．このための戦略の構築は今後の課題の一つである．

4.2　サイトのスコア計算

同じサンプル集合に対する DIPRE の繰り返し処理が収束する（つまり，このサンプル集合において新たなレコードが抽出できなくなる）まで処理を行うのは，同じ文書を多くの回数スキャンするため，効率な手法ではない．DIPRE の処理をどの繰り返しで終了し，新たな文書を対象とするか，カバー率の利用を考える．**カバー率**とは，収束まで処理を続けたとき抽出できると予想されるレコード数のうち，現時点すでに抽出されたレコード数の割合である．カバー率が一定の閾値（例えば 80%）になった時点で DIPRE を終了することが考えられる．ここで問題となるのは抽出可能なレコード数の予測である．そのため，ここでは capture-recapture モデル [5] を利用する．以下ではこのモデルについて述べる．

capture-recapture モデルは，元々野生生物の数を見積もるのに生物学者によって用いられたモデルである．基本的な考え方を紹介する．まず，ある程度の数の対象動物を捕らえた後，それらに印をつけた後リリースする．動物が元の環境に戻った後で，もう 1 回の捕獲を行う．2 回の捕獲において得られた動物の数を分析し，実際に存在している動物の数を見積もる．

レコード抽出の場合では，動物の代わりにレコードを捕らえると考える．捕獲する処理がレコード抽出に対応している．まず，ユーザから与えられた例示レコード集合を，2 つの集合 S_1, S_2 に分ける．S_1 をシード集合として，DIPRE を何回か（例えば 3 回）実行して，得られたレコード集合を R_1 とする．R_1 のレコードの数を n_1 とする．次に，S_2 をシード集合として，同じく何回か DIPRE を実行して，得られたレコード集合を R_2 とする．R_2 のレコードの数を n_2 とする．R_1 と R_2 に共通に含まれるレコードの集合を R_3 とし，R_3 のレコードの数を m で表す．実際に存在するレコードの数を N とすれば，以下の式が近似的に成り立つ．

$$m = \frac{n_1}{N} \times n_2 \tag{1}$$

すなわち，レコード集合 R_2 を情報抽出（捕獲）する際，R_1 において抽出されていたレコードを再び抽出してしまう確率が n_1/N である．よって，1 回目と 2 回目の情報抽出で重複するレコードの数 m は n_1/N に n_2 を掛けた値となる．これにより，実際に存在するレコード数の予測値は

$$N = \frac{n_1 \times n_2}{m} \tag{2}$$

となる．実際には，1 回目の情報抽出と 2 回目の情報抽出は独立でなく，また抽出されるレコードにも抽出されやすさの偏りなどがあることから，上の推定はあくまでもおおまかな近似である．

この考え方を拡張して，R_1, R_2, R_3 内のレコードをサイトごとに分けて，各サイトに含まれるレコードの数を予測できる．サイト W_i 内のレコード数の予測値を以下の式で計算する．

$$N(W_i) = \frac{n_1(W_i) \times n_2(W_i)}{m(W_i)} \tag{3}$$

この予測値を用いて，各サイト W_i のスコアを以下の式で計算する．

$$score(W_i) = \frac{N(W_i)}{\text{サンプル集合内の } W_i \text{のページ数}} \tag{4}$$

この値はあるサイトのページをアクセスしたページ数が多いほど低くなり，そのサイトから抽出できるレコード数の予測値が高いほど高くなるという性質がある．つまり，数少ないページをアクセスすることで，数多くのレコードを抽出できるサイトは良いサイトと考える．

4.3 ユーザからのフィードバックの利用

本手法では，ユーザが抽出されたレコードに対し適宜フィードバックを行うことを想定する．ユーザはレコード集合をブラウジングし，着目したレコードに対し，以下のどれに相当するかを判断する．

（1） 抽出されたレコードが適合である．

（2） 抽出されたレコードにノイズが含まれている：レコードの抽出パターン自体に問題があり，レコードとして不適切なデータが抽出された場合にあたる．

（3） 抽出されたレコードの自体は正しいが，トピックが不適合である：たとえば，データベース分野の本の著者とタイトル情報が求められているのに，ビジネス分野の本の著者とタイトルが含まれている場合などである．

（2）の場合には，ユーザからのフィードバックに応じて，DIPRE の抽出パターンの見直しを行う．例えば，抽出結果に問題があると指摘された場合，DIPRE 処理において，該当するレコードの抽出に用いられたパターンを，今後のレコード抽出に用いないように抹消する処理を行う．一方，(3) の場合にはそれらのレコードを含むウェブサイトのスコアを低くするよう，提案手法を拡張する必要がある．これも今後の課題の一つである．

4.4 データクリーニング処理の統合

前節において，ユーザによるフィードバックのアイデアについて述べた．提案手法の実際の活用においては，大量のノイズを含んだレコードが抽出されると考えられることから，ユーザの労力をいかに削減するかが重要となる．ユーザに提示する以前に，抽出レコード集合からノイズを含むデータを検出し，データ中のノイズ削減を行うことが考えられる．大量のデータから（半）自動的ノイズを取り除き，データマイニングの前処理を行ったり，データ統合の精度を向上させるための技術として，近年**データクリーニング** [18] が着目されている．

本手法では，特に前節で述べた（2）の項目に該当する，ノイズを含むデータの削減に対してデータクリーニング処理を統合することを検討している．

5. 議論と今後の課題

5.1 サイトスコアの他の計算手法の検討

サイトのスコアを計算するのに，類似度の利用も考えられる．以下ではこの方式のアイデアを簡単に述べる．レコード集合全体をテキストデータとみなし，ベクトル $\boldsymbol{q}$ を構成する．新たなレコードの追加が生じると，$\boldsymbol{q}$ も更新されることになる．サンプル集合に存在する各サイトの文書を単語のベクトル $\boldsymbol{d}$ で表現し，類似度 $sim(\boldsymbol{q}, \boldsymbol{d})$ をサイトのスコアとする．

もう一つの案として，あらかじめユーザレコードに関連するレコードを含むページを収集しておいて，分類器を作ることも考えられる．サンプル集合の各サイト内のページ集合を特徴ベクトルとみなし，分類器にかけて，スコアの高いサイトを優先的に今後の抽出用のリポジトリにする．これらの手法の詳細化は今後の課題である．

5.2 まとめと課題

本稿では，情報源の選択によりレコードを効率的に抽出する手法を提案した．抽出の結果をもとに，ウェブサイトが今後の抽出にどれだけ有用であるかを判断する．有用と判断されたサイトを優先的にレコードの抽出に利用することで，少ないコストで数多くのレコード抽出の実現を図る．今後は提案手法の詳細化と実験に基づく評価を行いたい．

謝　　辞

本研究の一部は，日本学術振興会科学研究費基盤研究 (C)(16500048)，同 (B)(15300027)，旭硝子財団研究助成，稲森財団研究助成及び CREST「自律連合型基盤システムの構築」による．

文　　献

[1] S. Brin, Extracting Patterns and Relations from the World Wide Web. *Proc. WebDB*, 1998.

[2] E. Agichtein and L. Gravano, Snowball: Extracting Relations from Large Plain-Text Collections. *Proc. ACM SIGMOD*, 2001.

[3] N. Kushmerick, Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, Vol. 118, No. 1-2, pp. 15–68, 2000.

[4] B. Liu, R. Grossman and Y. Zhai, Mining Data Records in Web Pages. *Proc. KDD*, 2003.

[5] R. Y. Zhang, L. V.S. Lakshmanan and R. H. Zamar, Extracting Relational Data from HTML Repositories. *SIGKDD Explorations*, 2004.

[6] D. A. Grossman and O. Frieder, *Information Retrieval: Algorithms and Heuristics*, Springer, 2004.

[7] D. Hawking and P. Thomas, Server Selection Methods in Hybrid Portal Search. *Proc. ACM SIGIR*, 2005.

[8] J. P. Callan, Z. Lu and W. Bruce Croft, Searching Distributed Collections with Inference Networks. *Proc. ACM SIGIR*, 1995.

[9] L. Gravano, Hector Garcia-Molena and A. Tomasic, GlOSS: Text-source Discovery over the Internet. *ACM Transaction on Database Systems*, 24(2), 1999.

[10] L. Si and J. Callan, Relevant Document Distribution Estimation Method for Resource Selection. *Proc. ACM SIGIR*, 2003.

[11] L. Si, R. Jin, J. Callan and P. Ogilvie, A Language Modeling Framework for Resource Selection and Results Merging. *Proc. CIKM*, 2002.

[12] metacrawler. `http://www.metacrawler.com/`

[13] search.com. `http://www.search.com/`

[14] P. G. Ipeirotis and Luis Gravano, Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection. *Proc. VLDB*, 2002.

[15] Z. Wu, W. Meng, C. Yu and Z. Li, Towards a Highly-scalable and Effective Metasearch Engine. *Proc. WWW*, pp. 386–395, 2001.

[16] E. Rahm and H.H. Do, Data Cleaning: Problems and Current Approaches. *Data Engineering Bulletin*, 23(4), 2000.

[17] H. Galhardas, D. Florescu, D. Shasha, E. Simon and C.-A. Saita, Declarative Data Cleaning: Language, Model, and Algorithms. *VLDB*, pp. 371–380, 2001.

[18] V. Raman and J.M. Hellerstein, Potter's Wheel: An Interactive Data Cleaning System. *Proc. VLDB*, pp. 381–390, 2001.