ブートストラッピングによるスキーマ抽出の基礎検討

大阪大学大学院　基礎工学研究科
野村 慎太郎　中根 史敬
土方 嘉徳　西田 正吾

背景(1/2)

Web上の半構造化文書

属性名

- メーカー型番 : 170957I
- タイプ : B5ノート
- CPU : Core 2 Duo
- メモリ : 1GB
- HDD : 120GB

属性値

データベースのように整形されておらず、データを計算機で扱えない

スキーマ(属性名の組)の抽出

属性名に対応する属性値の抽出

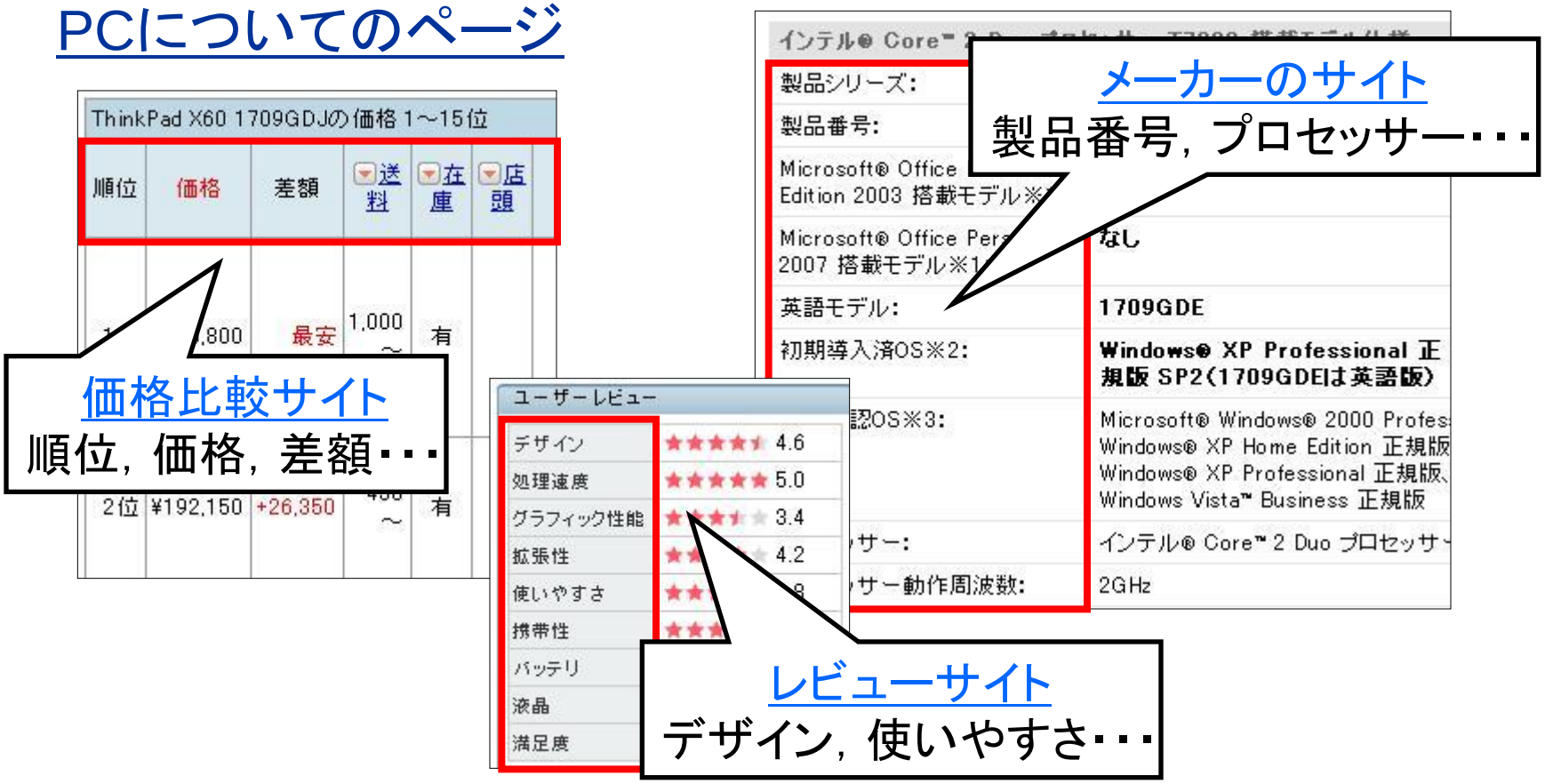
が必要

比較的多くの研究が行われてきた

ほとんど研究が行われていない

背景(2/2)

PCについてのページ



Webサイトの目的により、属性は大きく異なる

関連研究(1/2)

属性値抽出手法

- 人手で作成した抽出ルールを利用[Appelt 93][Baumgartner 01]
- 人手で抽出位置にタグ付けをしておき，機械学習によりルールを学習[Kushmerick 97][Ambite 98][山田 02]
- 少量の値を与えておき，ルールと抽出値を交互に繰り返し学習（ブートストラッピング）[Riloff 99][Yangarber 02][Ciravegna 04][楠村 07]

**多様な形式をもつWeb文書においても
ある程度の精度で抽出できる**

関連研究(2/2)

スキーマ抽出手法

- フォームに隣接するテキストを属性名として抽出
  [Raghavan 01][Zhang 04]
- 表の論理的構造を特定することにより属性名を抽出
  - 人手でセルにラベル付けした事例を用いる [Tengli 04]
  - セル中のテキスト間の類似度を用いる [Chen 00]
- クラス語との共起パターン，共起頻度から属性名を特定
  [Tokunaga 05]

本研究の目的

既存のスキーマ抽出手法

- 特定の形式(フォーム、テーブル)のみが対象
- 人手によるラベル付けが必要

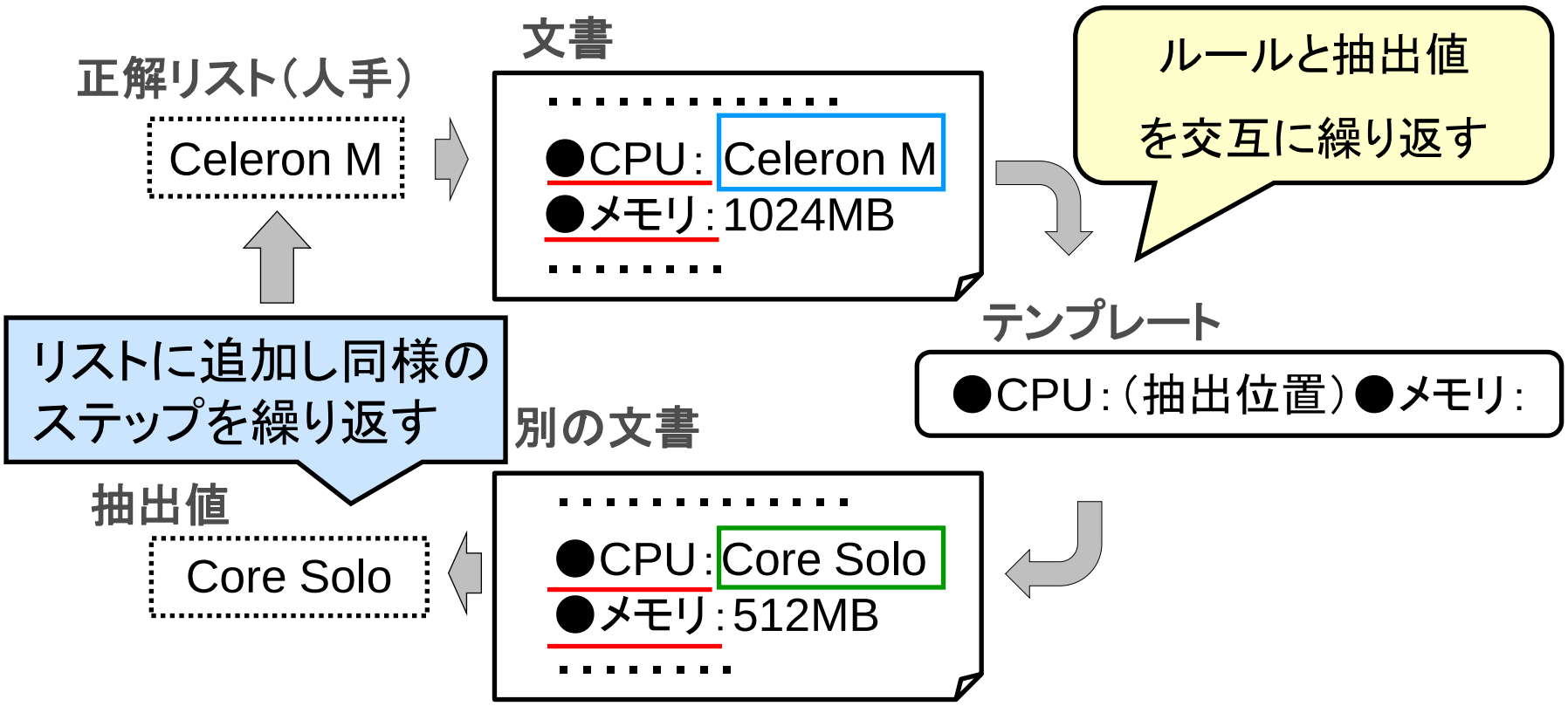
Web ページにおける記述の多様性に対応し、
網羅的にスキーマを抽出できない

本研究

ブートストラッピングによりスキーマの自動抽出を目指す

ブートストラッピング

人手で属性値を少数与えておき、抽出値とテンプレートを繰り返し学習することで大量の属性値を得る手法



属性名抽出における問題点

属性値抽出の場合

<LI><B>CPU：</B>Celeron M

<LI><B>メモリ：</B>1024MB

属性名抽出の場合

<LI><B>CPU：</B>Celeron M

<LI><B>メモリ：</B>1024MB

——：テンプレート

テンプレートに**属性名**が含まれ，多様性は低い

テンプレートに**属性値**が含まれ，多様性が高い

適合する確率が低くなる

属性名抽出の提案方法

2つの属性名の前後にあるテキストで、
完全一致する部分をテンプレートとする

検索語の例

『CPU』『メモリ』

```
<DIV class=content  id=product>
<UL>
<LI> <B> メーカー型番 </B> 170957I
<LI> <B> タイプ </B> B5ノート
<LI> <B> CPU </B> Core 2 Duo
<LI> <B> メモリ </B> 1GB
<LI> <B> HDD </B> 120GB </UL> </DIV>
```

属性名、属性値が
書かれる構造は
同じであることが多い

テンプレート

<LI><B>（抽出位置）:</B>

テンプレートをWeb全体には適用せず、
現在のページにのみ適用して属性名を抽出する

スキーマ抽出の問題点

(1)用語解説のようなページも検索に適合してしまう

(2)多様なスキーマを抽出できない

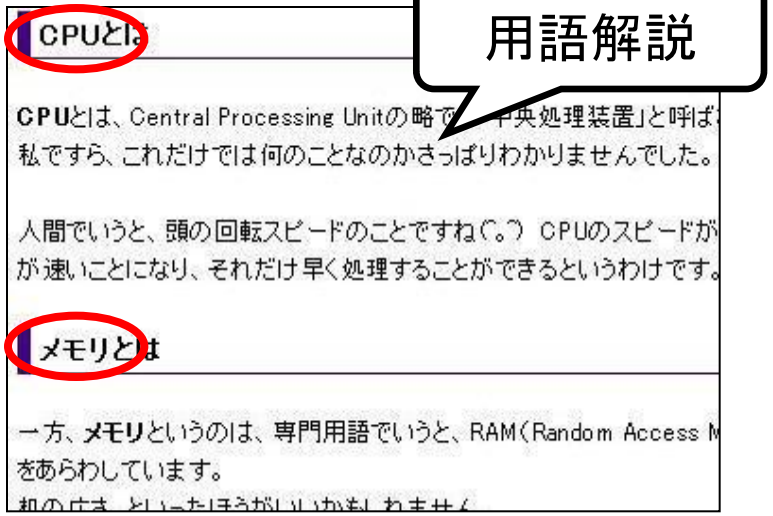
(3)テンプレートがページ中の様々な位置に適合する可能性がある

問題点と解決方法(1/3)

2つの属性名によるページ検索

検索語

『CPU』『メモリ』



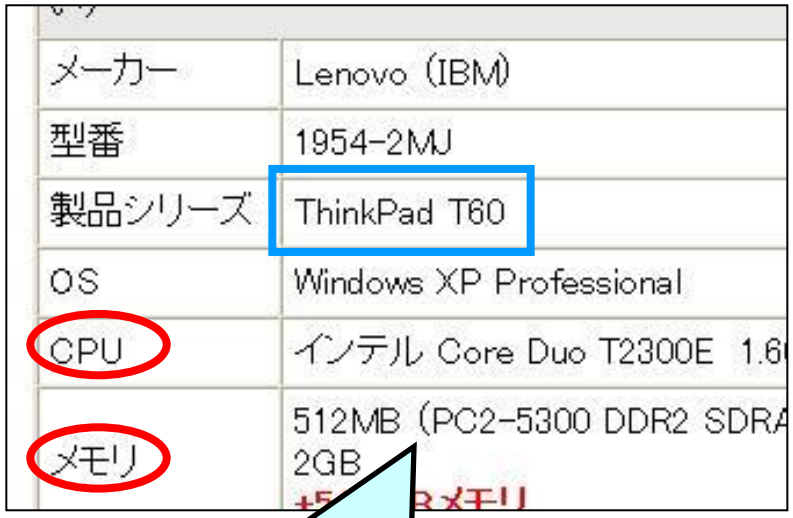
必要でないページ

必要なページ

用語解説のようなページも検索に適合してしまう

➡ **『属性名＋キー』でページを検索**

問題点と解決方法(2/3)



メーカー	Lenovo (IBM)
型番	1954-2MJ
製品シリーズ	ThinkPad T60
OS	Windows XP Professional
CPU	インテル Core Duo T2300E 1.6
メモリ	512MB (PC2-5300 DDR2 SDRA 2GB

Let's note LIGHT R7
本体サイズ: B5相当
CPU: Core 2 Duo
メモリ: 1GB
画面サイズ: 10.4インチ
画面解像度: 1024x768
HDD: 120GB
対応: 無線LAN

新たな属性名

メーカー，型番，
製品シリーズ，OS

本体サイズ，画面サイズ，
画面解像度，HDD・対応

異なるキーを用いると多様なスキーマを獲得できる

➡ キーもブートストラッピングで獲得

問題点と解決方法(3/3)

テンプレートがページ中の様々な位置に適合する可能性がある

テンプレート

<LI><B>(抽出位置):</B>

タグや記号だけから
構成され，短い

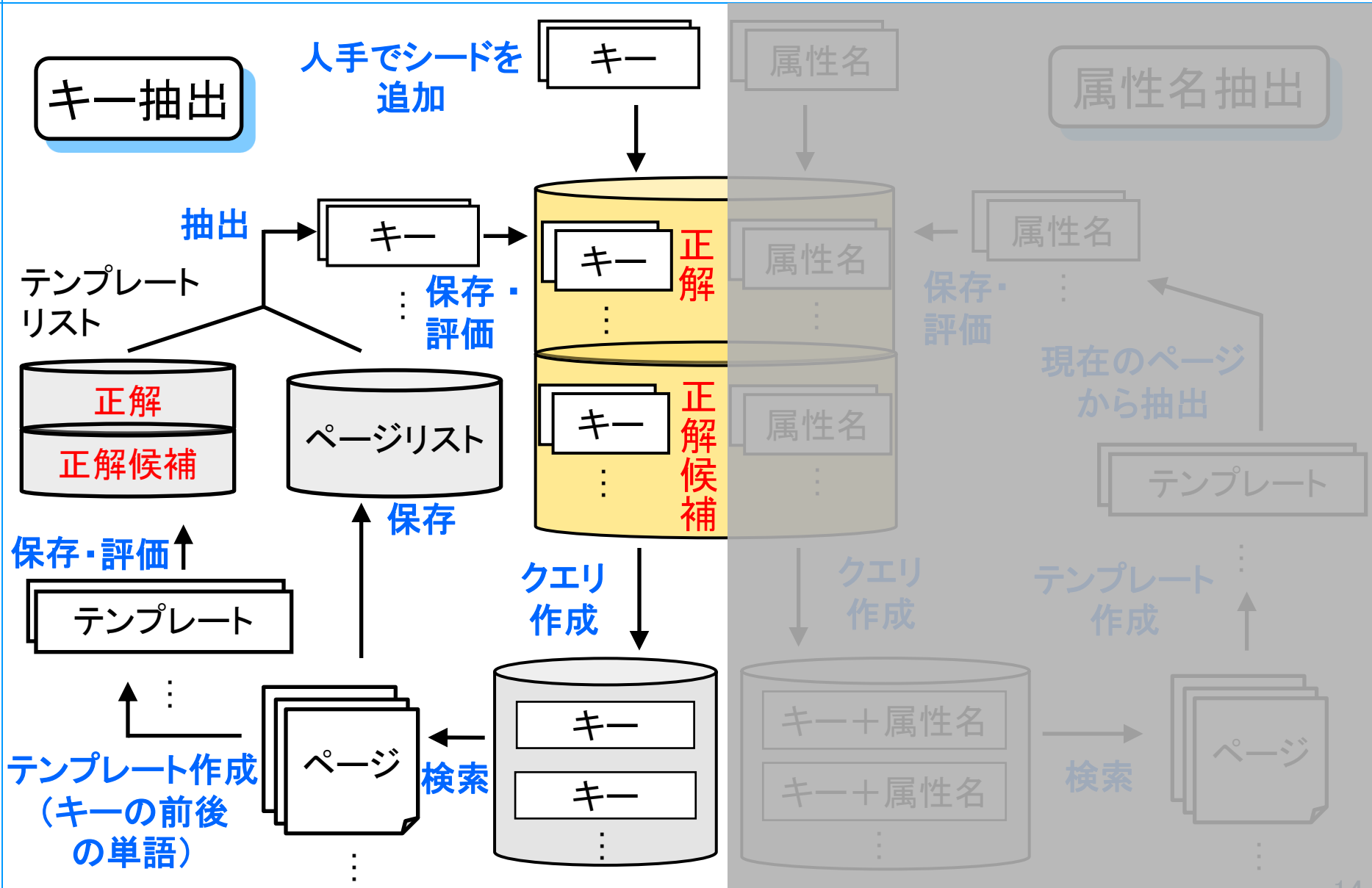
<DIV class=content id=product>
<UL>
<LI><B>メーカ
<LI><B>タイプ
<LI><B>CPU:</B>Core 2 Duo
<LI><B>メモリ:</B>1GB
<LI><B>HDD:</B>120GB</UL></DIV>
:
<LI><B>関連商品:</B> ・・・・
<LI><B>お問い合わせ:</B>・・・・

この部分だけから抽出したい

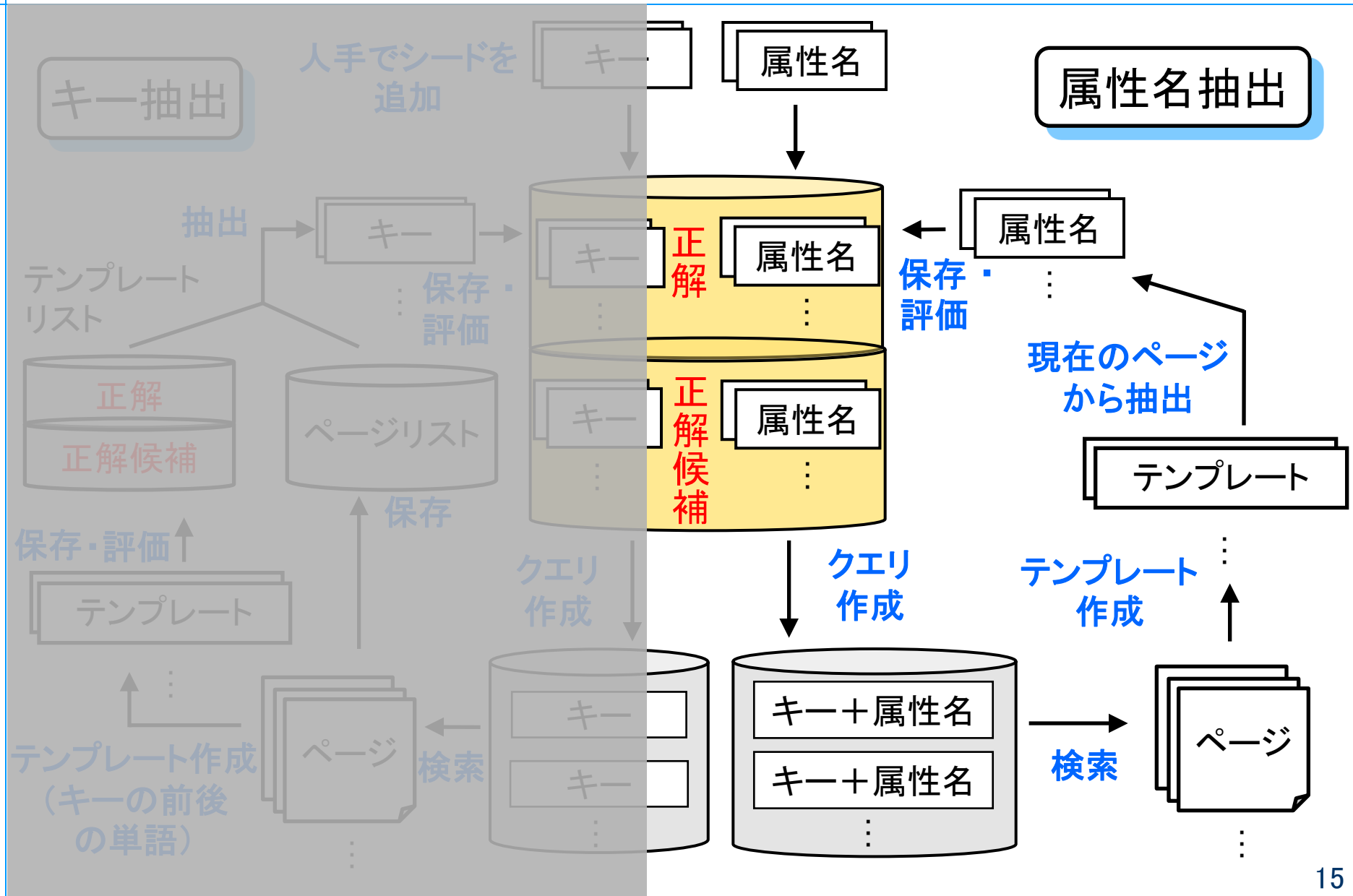
これらにも適合

文書構造を参照し，抽出を行う範囲を限定

キー抽出処理の流れ


キー抽出
人手でシードを追加
キー
抽出
キー
テンプレートリスト
保存・評価
キー
正解
キー
正解候補
正解
正解候補
ページリスト
保存
保存・評価
テンプレート
クエリ作成
キー
キー
ページ
検索
テンプレート作成（キーの前後の単語）
属性名抽出
属性名
属性名
属性名
保存・評価
現在のページから抽出
テンプレート
テンプレート作成
クエリ作成
キー＋属性名
キー＋属性名
検索
ページ

属性名抽出処理の流れ


キー抽出
人手でシードを追加
キー
属性名
属性名抽出
抽出
キー
テンプレートリスト
保存・評価
正解
正解候補
ページリスト
保存
保存・評価
テンプレート
テンプレート作成（キーの前後の単語）
ページ
検索
クエリ作成
正解
属性名
正解候補
属性名
保存・評価
属性名
現在のページから抽出
テンプレート
クエリ作成
テンプレート作成
キー＋属性名
キー＋属性名
検索
ページ

キー抽出実験

目的 精度・抽出数が最も良くなるキー抽出パラメータの調査

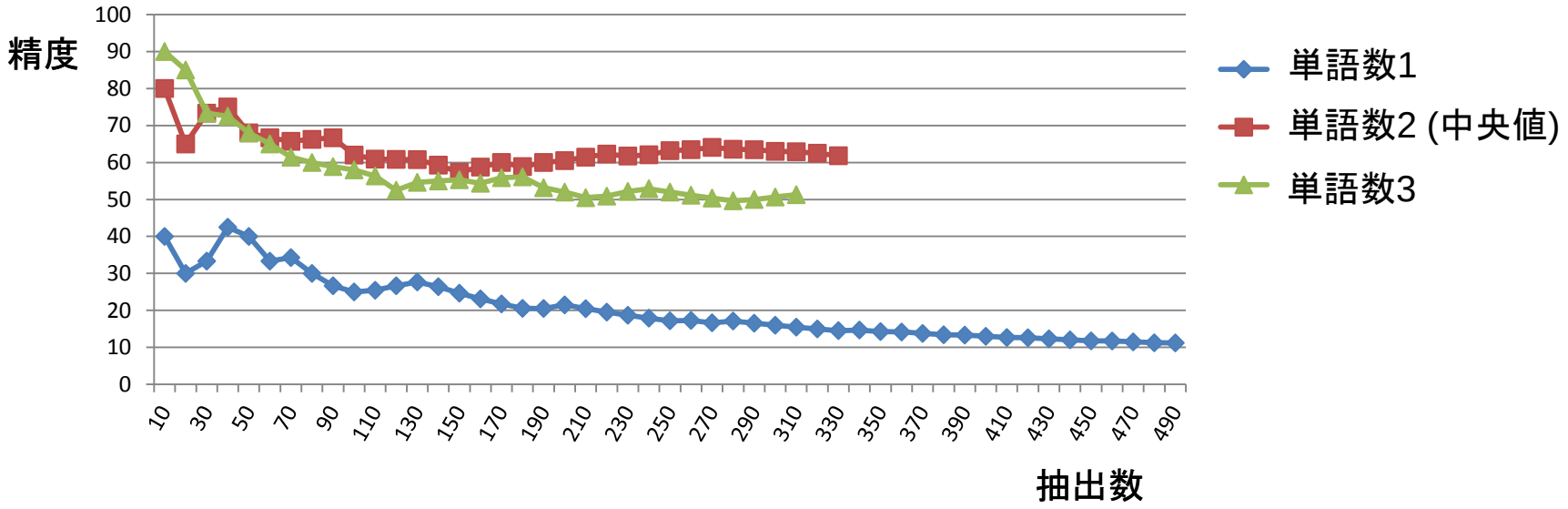
- テンプレート単語数：テンプレートに用いる単語の数
- キー更新数：各サイクルで正解とするキーの数
- テンプレート更新数：各サイクルで正解とするテンプレートの数

パラメータ	各パラメータの値 （中央値）					
テンプレート単語数	1	2	3	4	5	6
キー更新数	1	2	5	10	20	50
テンプレート更新数	10	20	50	100	200	500

1つのパラメータを変化させて実験を行う
他2つのパラメータは中央値をとる

結果(1)

テンプレート単語数を変化させた場合の精度の推移



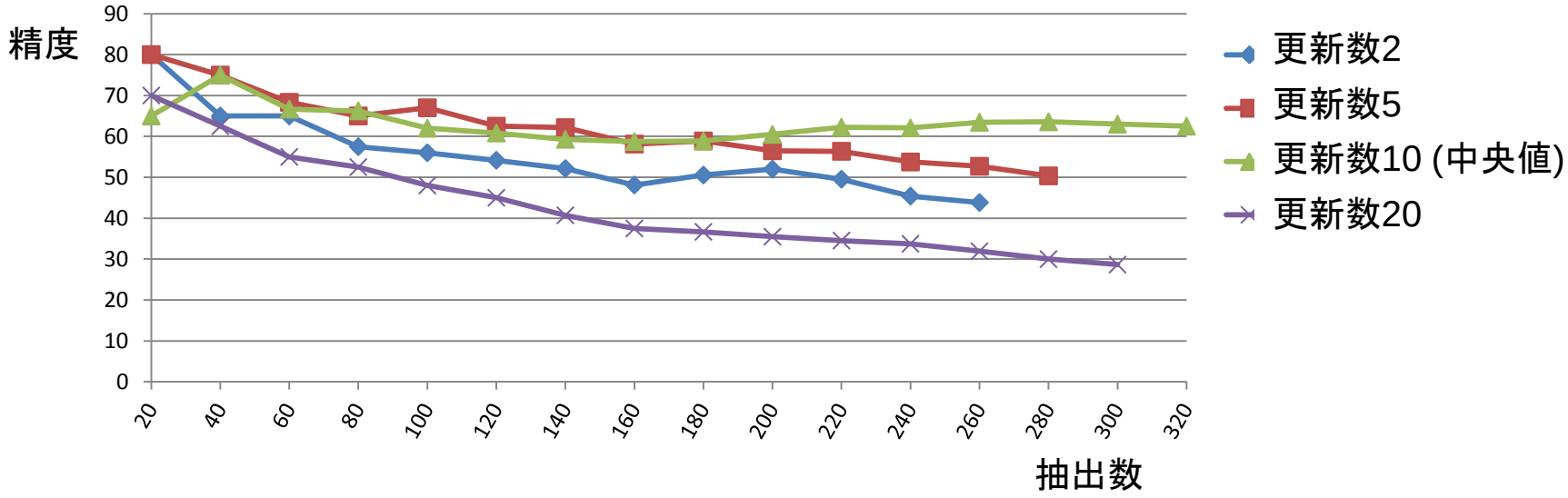
- 抽出数が30までは、単語数=3が良い
- 5万ページ取得時には、単語数=2が良い

パラメータ:テンプレート単語数
サイクル:初期3、以降2

が良い

結果(2)

キー更新数を変化させた場合の精度の推移



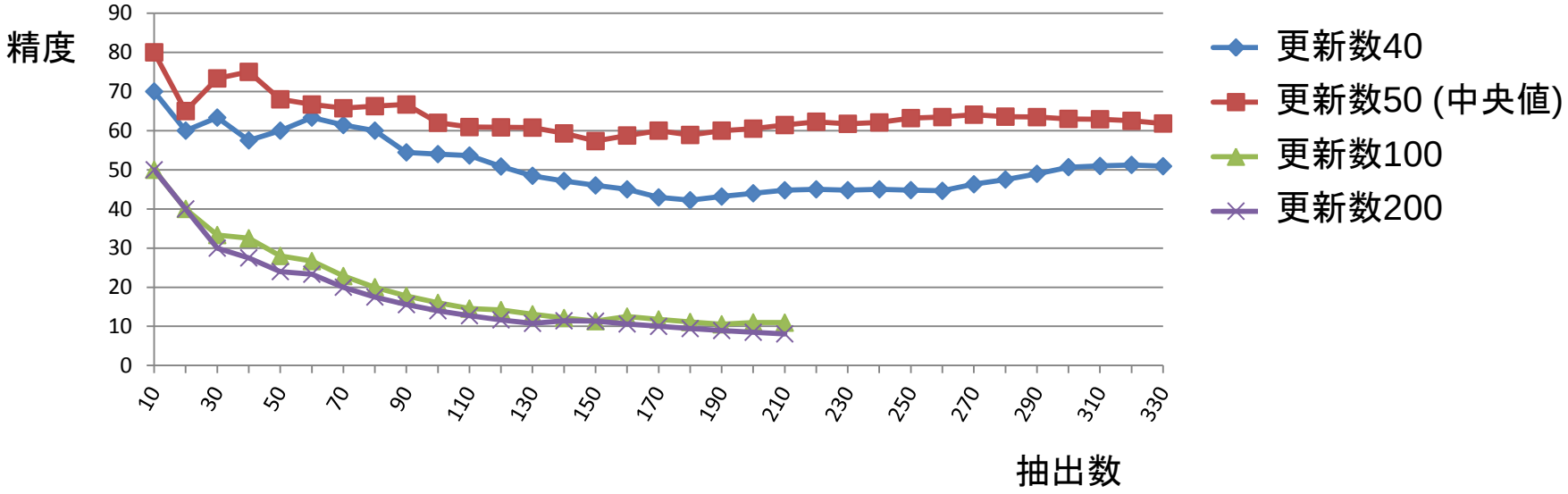
- サイクルの前半では、更新数=5が最も精度が良い
- サイクルの後半では、更新数=10が最も精度が良い

パラメータ：キー更新数
サイクル：前半5、後半10

が良い

結果(3)

テンプレート更新数を変化させた場合の精度の推移



- 精度は更新数が 50>40>100>200 の順に良い

パラメータ:テンプレート更新数
サイクル:終始50

が良い

まとめ

- ブートストラッピングを用いたスキーマ抽出手法の提案
  - キー＋2つの属性名で検索し，前後テキストの完全一致部分からテンプレートを作成
  - 属性名とキーをそれぞれブートストラッピングで獲得
  - 文書構造を参照し、抽出を行う範囲を限定

- キー抽出パラメータと精度・抽出数の関係を5万ページで調査
  結果として、パタメータ固定の場合、(2、10、50)が最良

- 今後の予定:
  途中再開のプラグラムを追加し、実験の開始