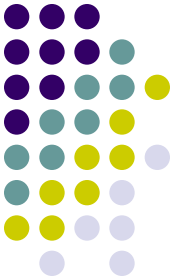
日本語解析ツール MeCab, CaboCha の紹介

工藤 拓

形態素解析とは

- 文を単語に区切り品詞を同定する処理
  - 明示的な単語境界が無い言語では必須の処理
  - 全文検索　Spam フィルタリング　人工無能...
- 以下の3つの処理
  - 単語への分かち書き(tokenization)
  - 活用語処理(stemming, lemmatization)
  - 品詞同定(part-of-speech tagging)
- MeCab:
  - 汎用テキスト処理フレームワーク
  - 形態素解析**も**できる
  - かな漢字変換等にも応用可能

MeCabの採用実績



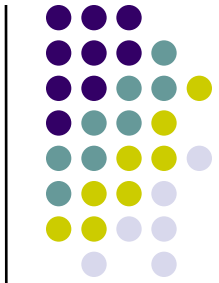
- MacOSX Leopard に標準搭載
  - Spotlight (全文検索) のインデックスに利用
- iPhone のかな漢字変換エンジン
- 全文検索エンジンのインデクシング
  - Senna
  - Hyper Estraier
  - Lucene
- Google Japanese n-gram
  - Web ページの 200億文の解析

MeCab の出力

```
私は新しく建った図書館へ行った
私　　名詞,代名詞,一般,*,*,*,私,ワタシ,ワタシ
は　　助詞,係助詞,*,*,*,*,は,ハ,ワ
新しく　形容詞,自立,*,*,形容詞・イ段,連用テ接続,新しい,アタラシク,アタラシク
建っ　　動詞,自立,*,*,五段・タ行,連用タ接続,建つ,タッ,タッ
た　　助動詞,*,*,*,特殊・タ,基本形,た,タ,タ
図書館　名詞,一般,*,*,*,*,図書館,トショカン,トショカン
へ　　助詞,格助詞,一般,*,*,*,へ,ヘ,エ
行っ　　動詞,自立,*,*,五段・カ行促音便,連用タ接続,行く,イッ,イッ
た　　助動詞,*,*,*,特殊・タ,基本形,た,タ,タ
EOS
```

- 単語 <tab> 品詞1,品詞2,品詞3,品詞4,活用型,活用系,基本形,読み,発音
- EOS: End of sentence
- 自由にフォーマット変更可能 (演習で説明します)

形態素解析の技術



- 基本的な処理: 辞書から単語を引いて、与えられた文と照合し、最も自然な単語列を求める
  - 辞書引き
    - 入力文は単語毎に区切られていない
    - どの文字列を辞書引きするか自明ではない
  - 曖昧性の解消
    - すべての可能な単語の組合せから(何らかの基準で)最適な単語列を発見する
    - 基準の定義

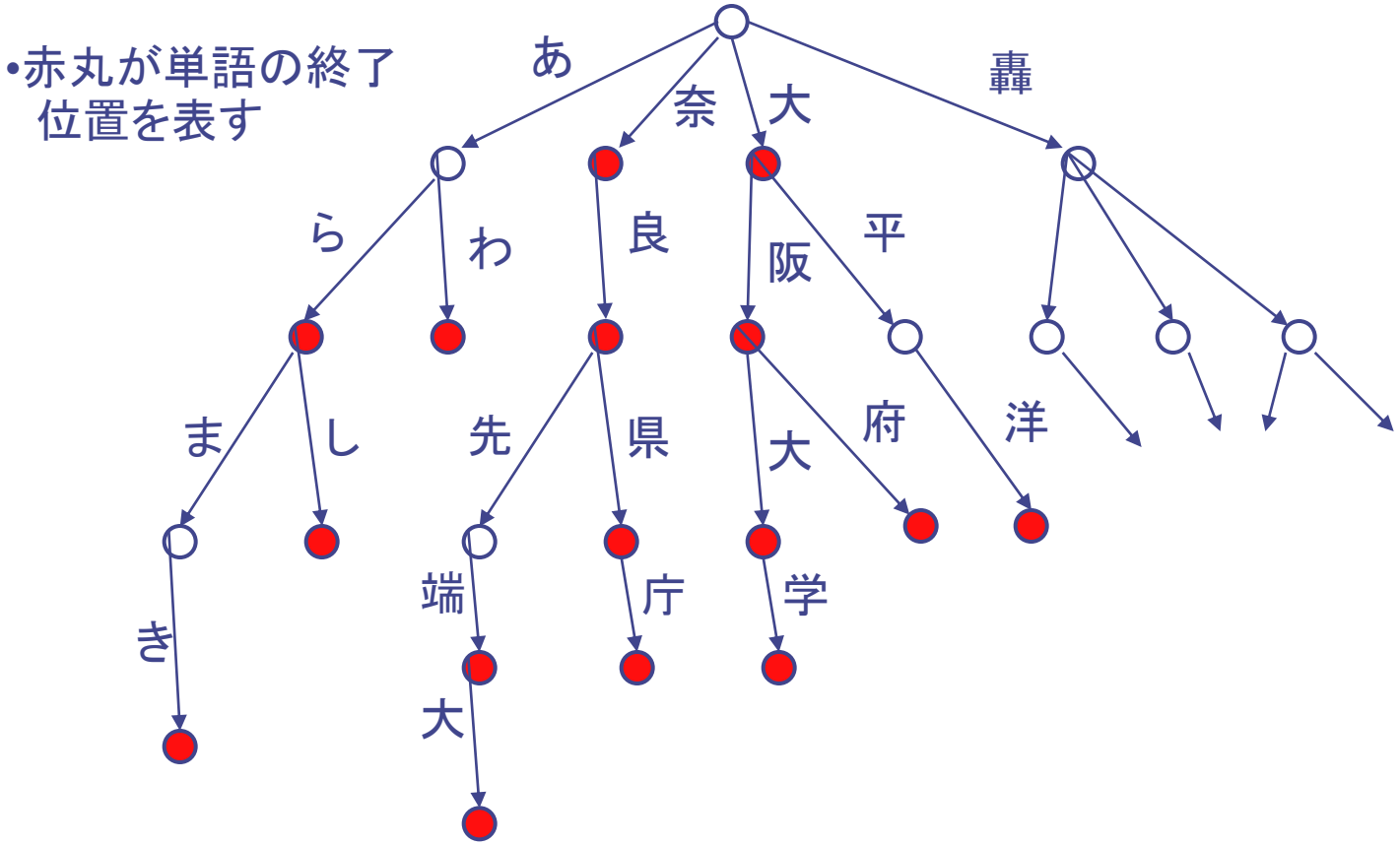
日本語処理のための辞書の要件

- 単語の区切りが明確でないので、先頭から何文字までが単語なのかわからない

奈良先端科学技術大学院大学情報科学研究科

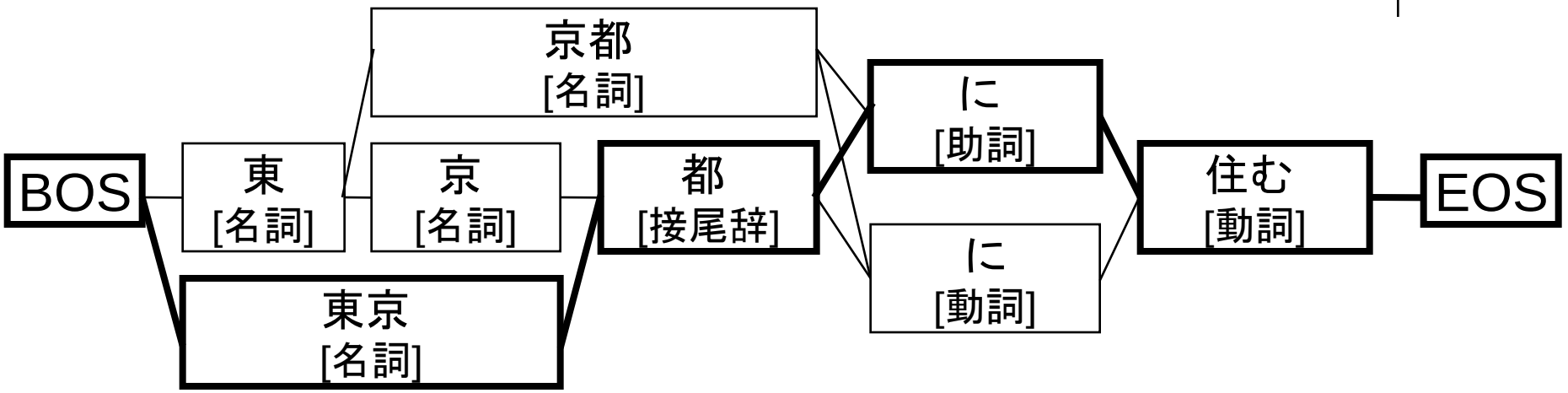
- しらみつぶしの方法だと (文長)^2 の辞書引きが発生
  - リレーショナルデータベース等は使えない
  - 日本語処理に特化したデータ構造が必要

辞書検索のためのデータ構造：TRIE



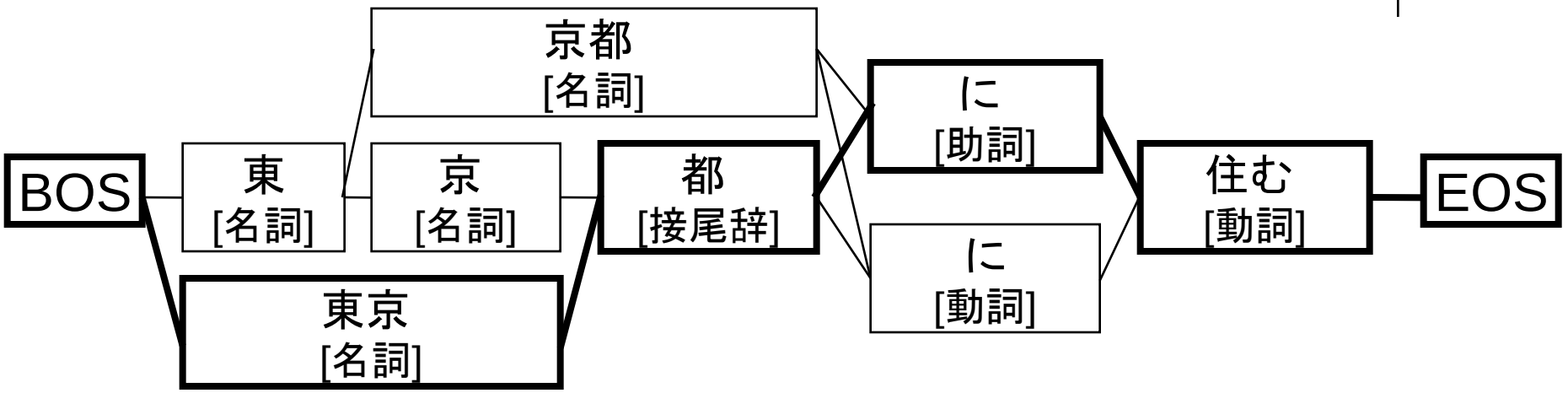
- 対象文字列の先頭から文字を順番にたどるだけ
- 辞書引き終了のタイミングが自動的にわかる
- さまざまな実装, MeCab は ダブル配列を使用

形態素ラティスと問題設定



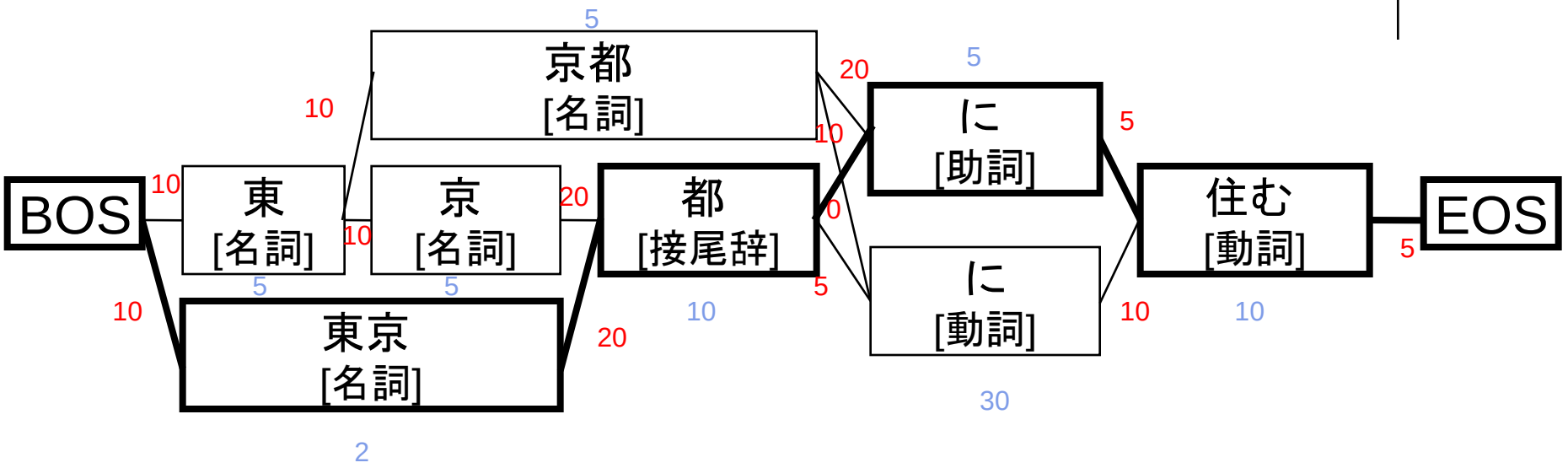
- 形態素ラティス
  - すべての可能な出力をグラフで表現した構造
  - 辞書引きをしながら構築
- 形態素解析の工学的な定義
  - 形態素ラティスから確からしい経路を見つける問題

経路探索の基準



- 規則(ヒューリスティックス)に基づく手法 (80年代)
  - 最長一致:　長い単語を優先 (KAKASI)
  - 分割数最小: 文全体の単語の数を最小にする候補
  - 文節数最小: 文全体の文節数を最小にする候補
- 多くの場合曖昧性を解決できない

最小コスト法

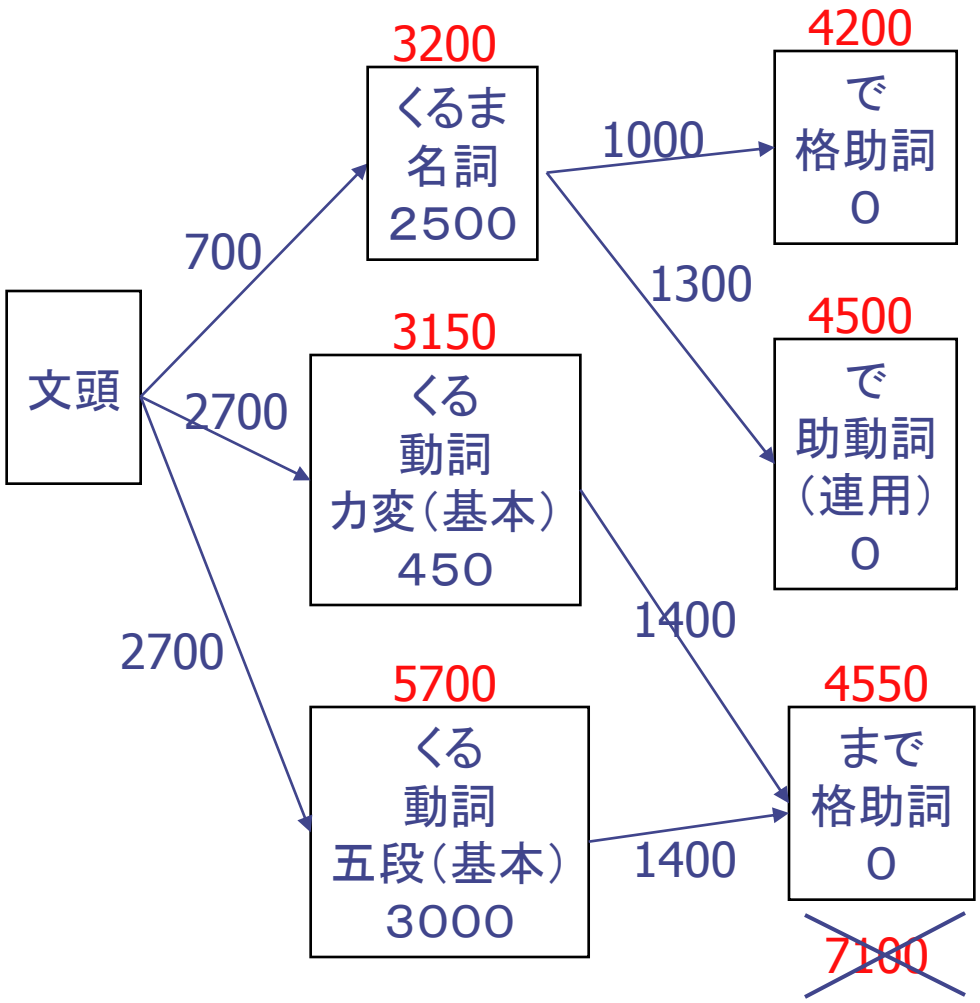


連接コスト: 二つの単語のつながりやすさ

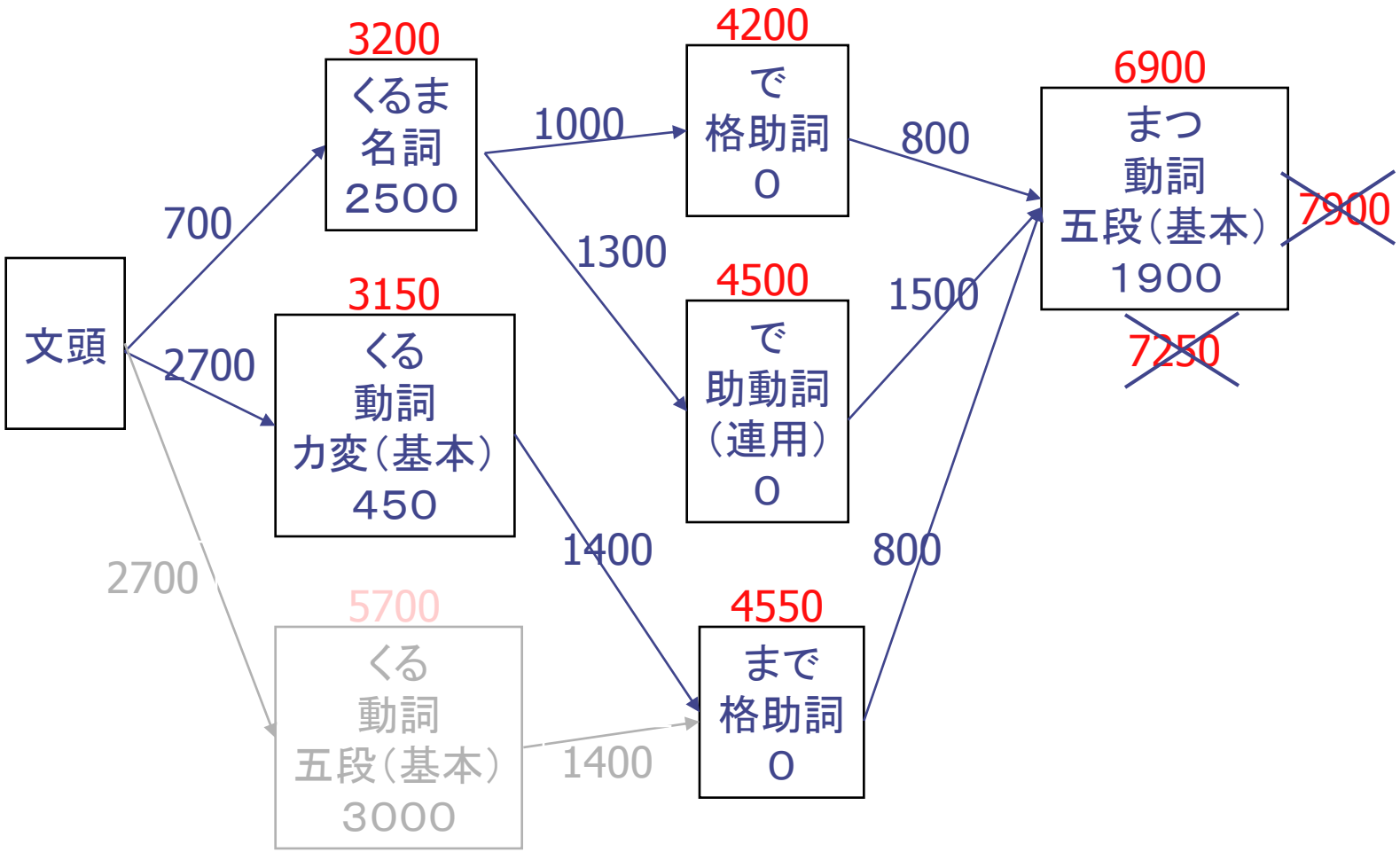
生起コスト: 一つの単語の出現しやすさ

- 連接コストと生成コストの和が最小になる解
- コストはなんらかの方法で決定 (後述)
- Viterbi アルゴリズムを使い最適解を求める

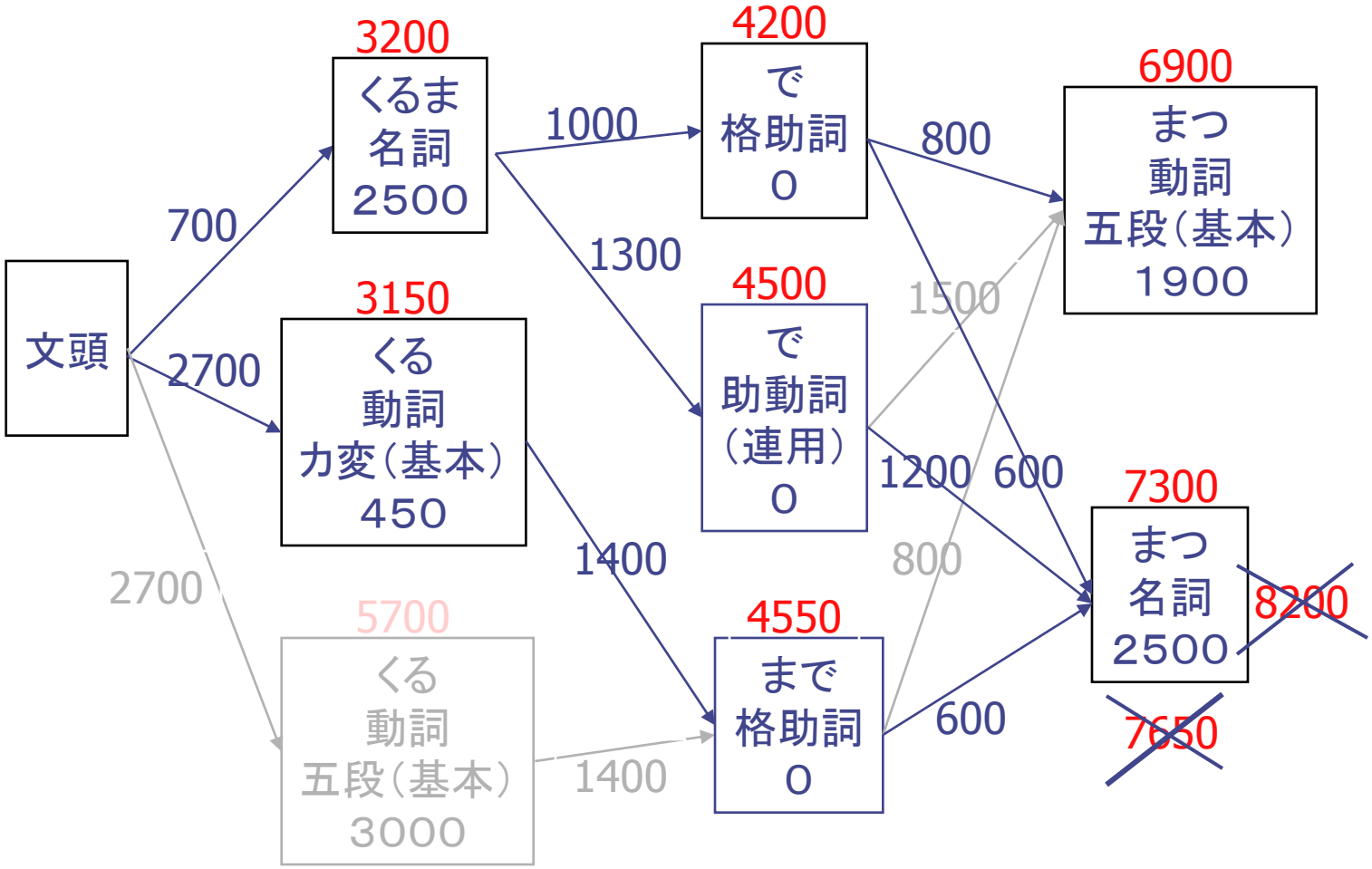
最小コスト法 (Viterbi アルゴリズム)



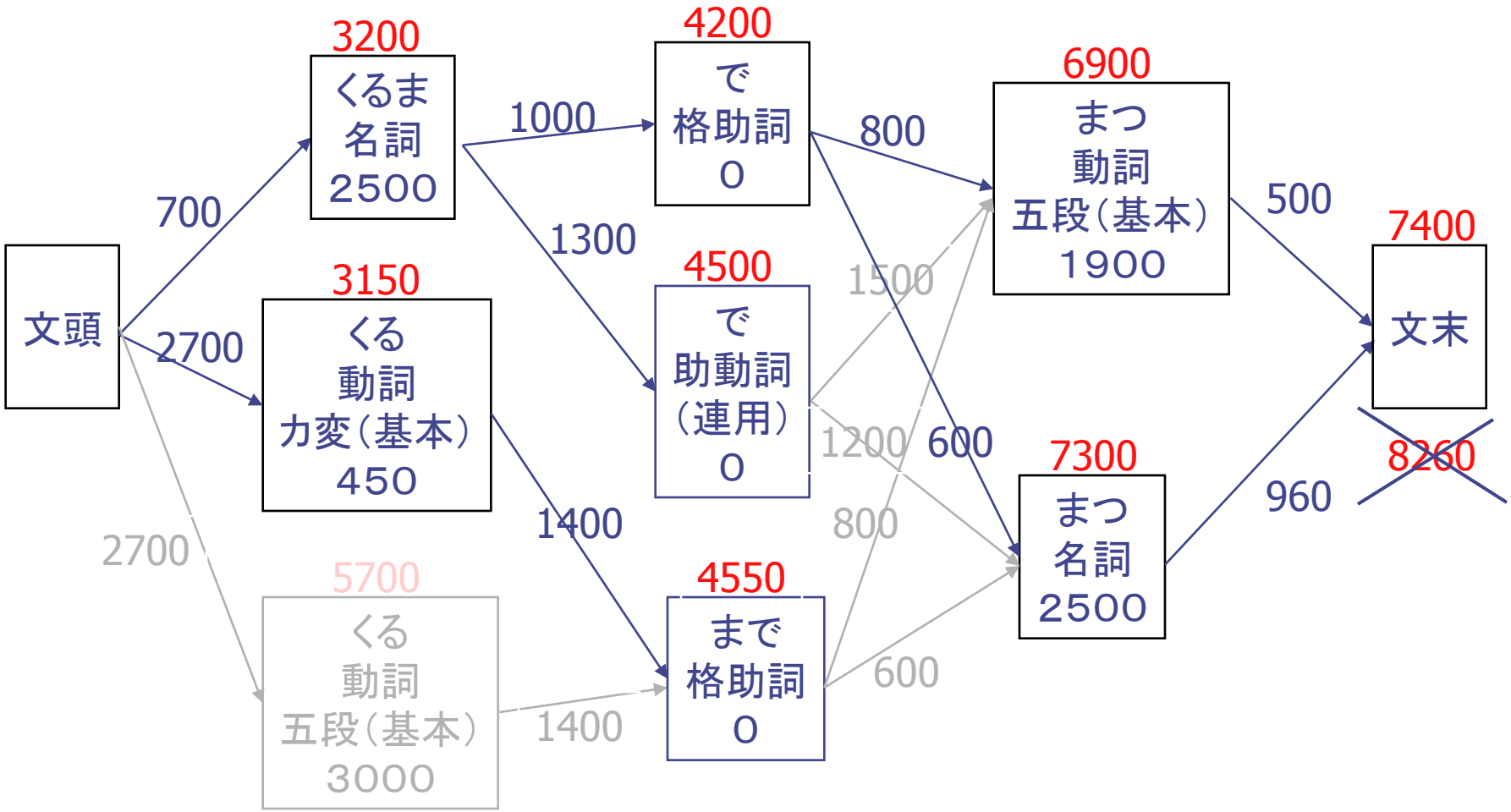
最小コスト法 (Viterbi アルゴリズム)



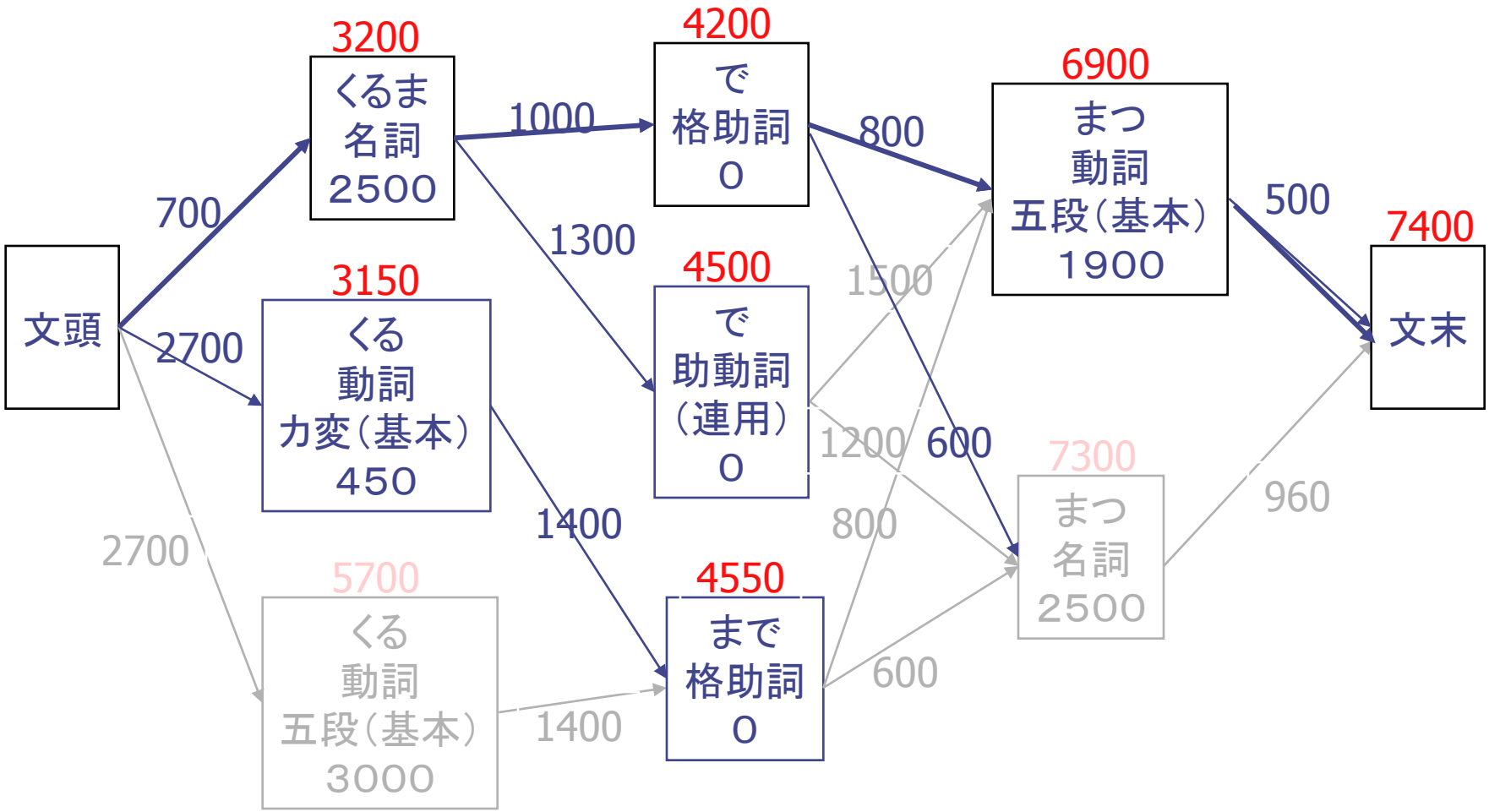
最小コスト法 (Viterbi アルゴリズム)



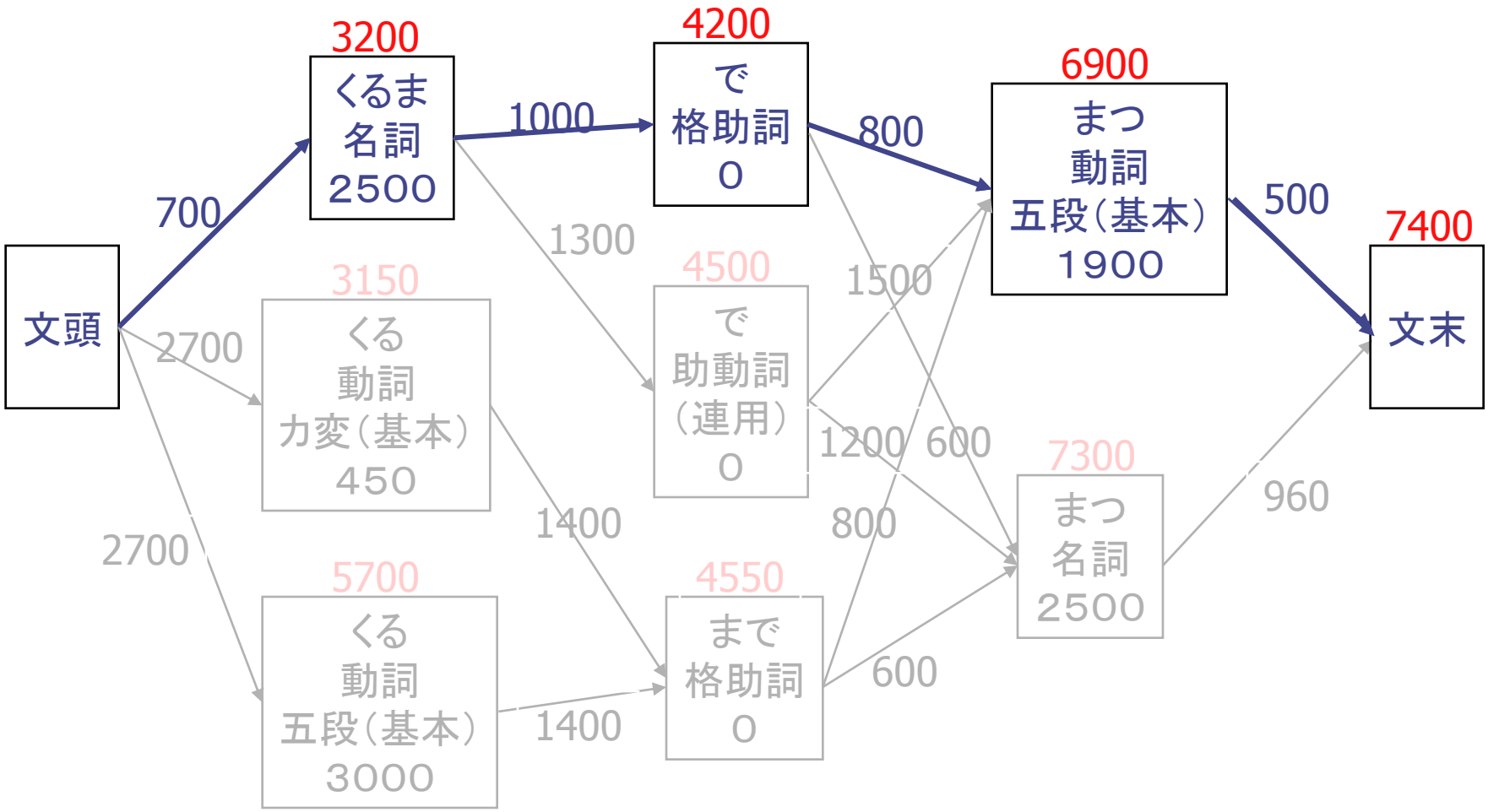
最小コスト法 (Viterbi アルゴリズム)



最小コスト法 (Viterbi アルゴリズム)



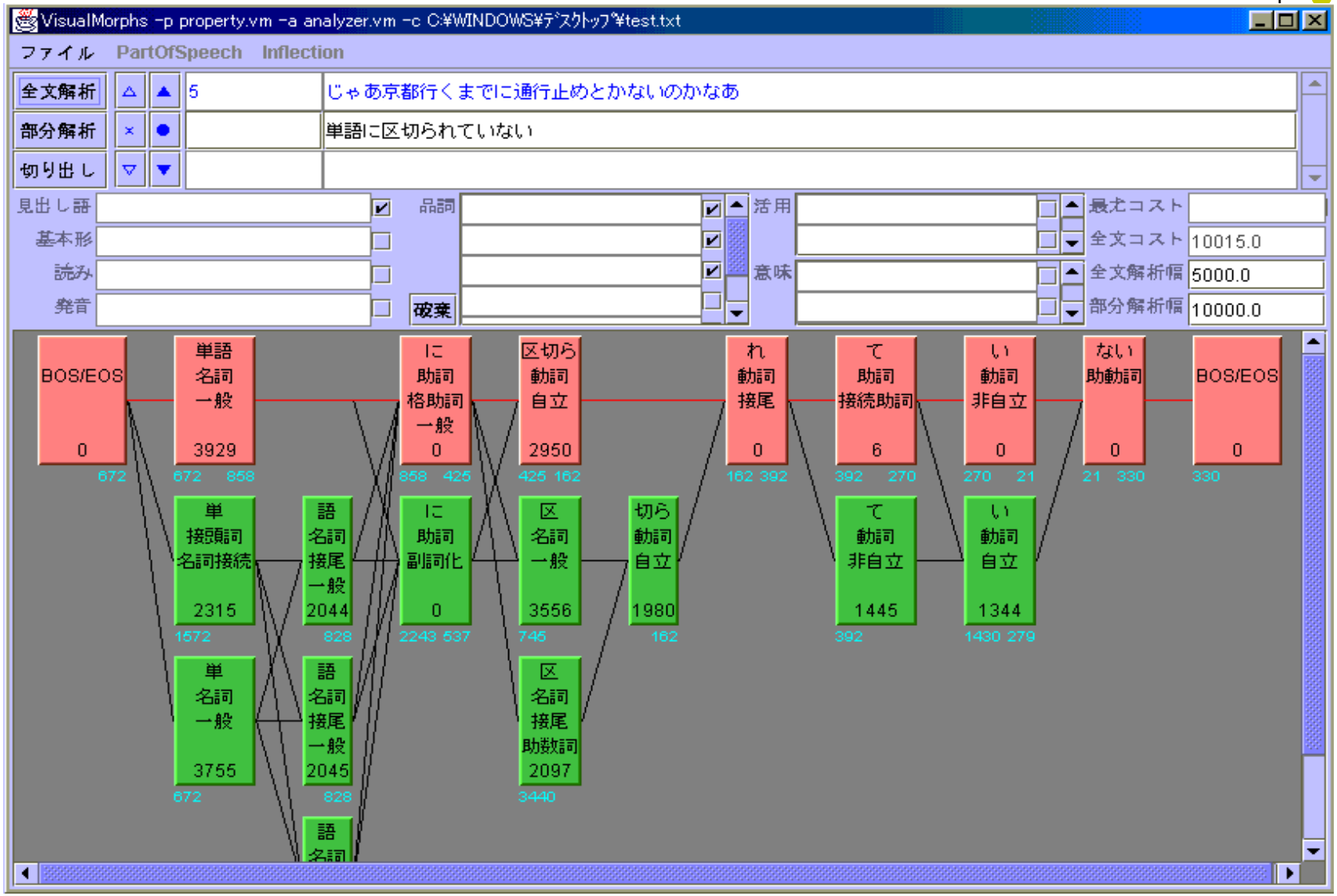
最小コスト法 (Viterbi アルゴリズム)



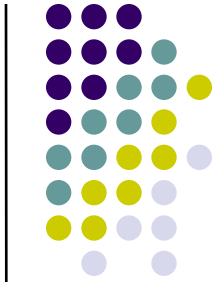
コストの決定方法

- 人手でガンバル (90年代はじめ)
  - 試行錯誤の連続
  - 客観的評価が難しい
- 統計処理
  - 大量の生テキストから推定
    - 質に問題がある (全文検索目的だったら可能かも)
  - 正解データを人手で作ってデータから推定
    - 現代の形態素解析器の主流
  - これから…
    - 大量の生テキスト + 少量の正解データ + 統計処理

正解データ作成ツール (VisualMorphs)

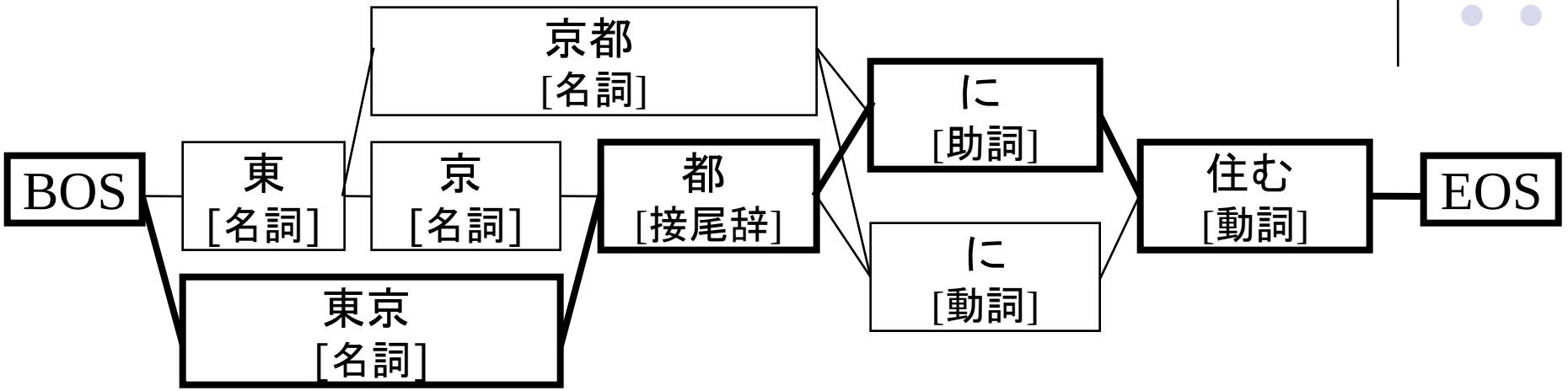


Conditional Random Fields



- MeCab 0.90 から採用された機械学習を用いたコスト推定アルゴリズム
- CRF 以前は HMM が使われていた
- HMMに比べて高性能
  - 1/3 程度の正解データで同程度の性能
- コスト推定モジュールを同封
- CRFの基本的な考え方
  - 正解が再現されるようなコスト値を探索
  - 人間が行なう試行錯誤を機械にやらせるイメージ

CRFによるコスト推定



連接コストテーブル

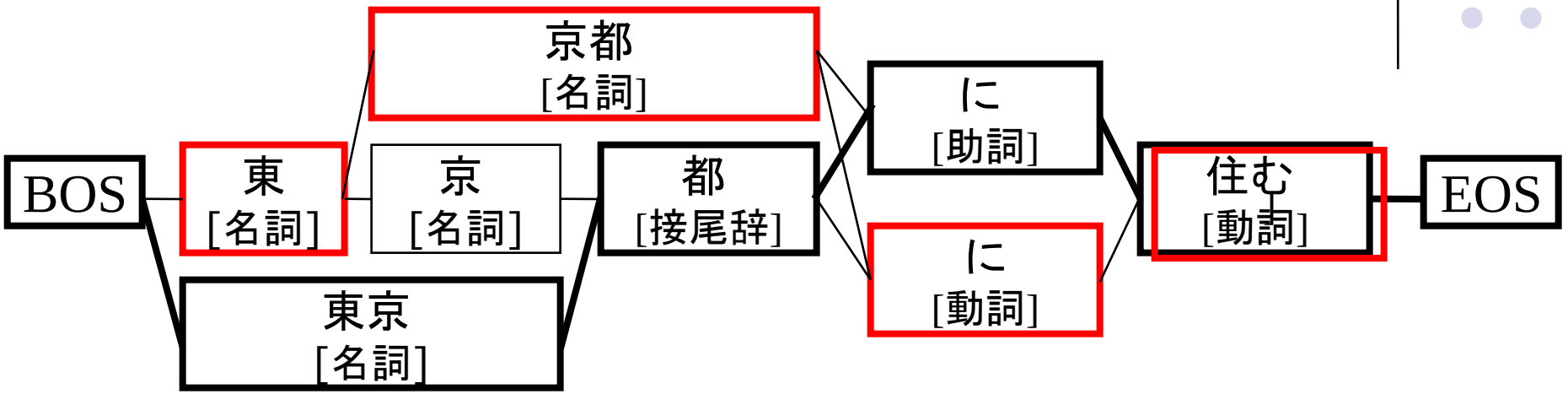
名詞-名詞: 0
名詞-助詞: 0
名詞-接尾辞: 0
名詞-動詞: 0
動詞-動詞: 0
接尾辞-助詞: 0
…

単語生起テーブル

名詞-東: 0
名詞-京都: 0
名詞-東: 0
動詞-住む: 0
接尾辞-都: 0
動詞-に: 0
…

- 太枠が正解の形態素解析結果 (人手で作成)
- 学習前は、全コストを 0 に初期化しておく

CRFによるコスト推定



連接コストテーブル

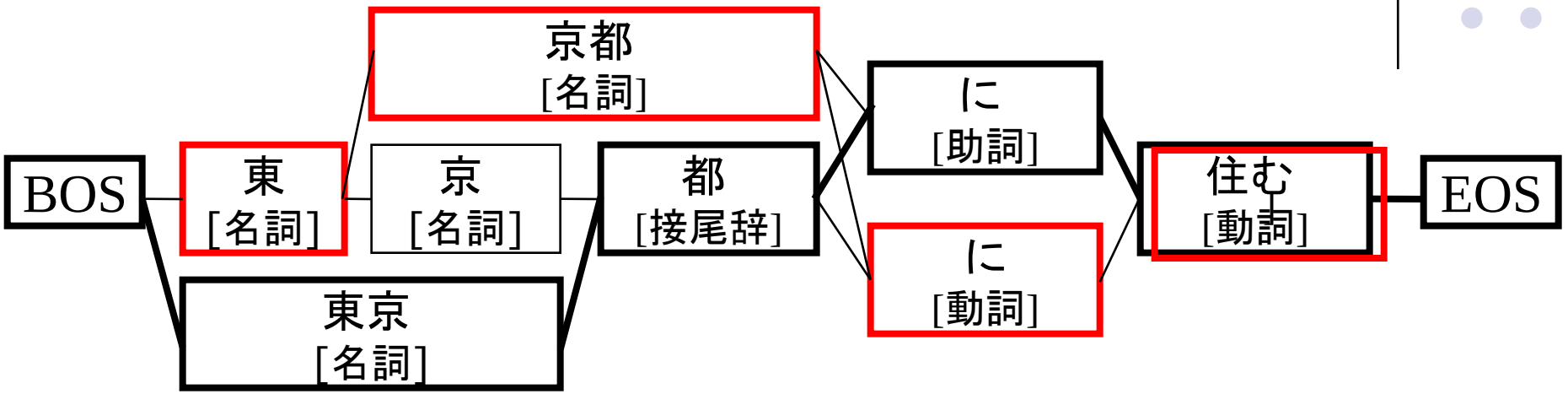
名詞-名詞: 0
名詞-助詞: 0
名詞-接尾辞: 0
名詞-動詞: 0
動詞-動詞: 0
接尾辞-助詞: 0
…

単語生起テーブル

名詞-東: 0
名詞-京都: 0
名詞-東: 0
動詞-住む: 0
接尾辞-都: 0
動詞-に: 0
…

- **赤枠**は現在のコストでの解析結果
- **赤枠**　コストを +1
- **黒枠**　コストを -1
- 正解のコスト値を小さくしていく

CRFによるコスト推定



連接コストテーブル

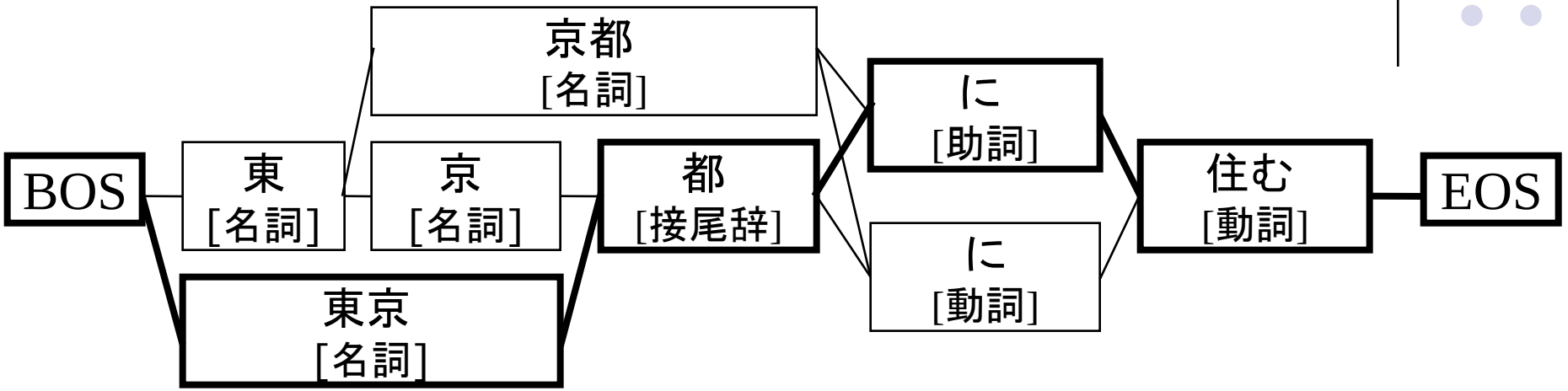
名詞-名詞: +1
名詞-助詞: 0
名詞-接尾辞: -1
名詞-動詞: 0
動詞-動詞: +1
接尾辞-助詞: -1
…

単語生起テーブル

名詞-東: +1
名詞-京都: +1
名詞-東: -1
動詞-住む: -1
接尾辞-都: -1
動詞-に: +1
…

- **赤枠**は現在のコストでの解析結果
- **赤枠**　コストを +1
- **黒枠**　コストを -1
- 正解のコスト値を小さくしていく

CRFによるコスト推定



連接コストテーブル

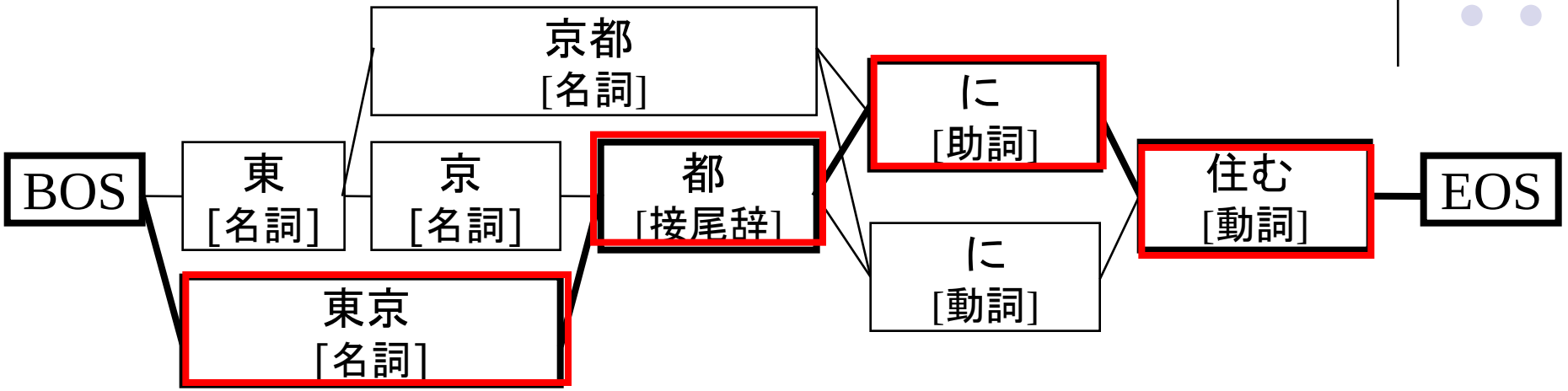
名詞-名詞: 1
名詞-助詞: 0
名詞-接尾辞: -1
名詞-動詞: 0
動詞-動詞: 1
接尾辞-助詞: -
…

単語生起テーブル

名詞-東: 1
名詞-京都: 1
名詞-東: -1
動詞-住む: -1
接尾辞-都: -1
動詞-に: 1
…

- 全学習データを1つずつ読み込み、コストを更新
- このステップを繰り返す

CRFによるコスト推定



連接コストテーブル

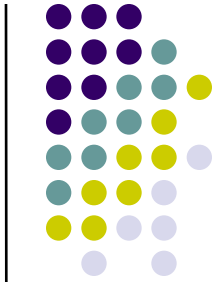
名詞-名詞: 1
名詞-助詞: 0
名詞-接尾辞: -1
名詞-動詞: 0
動詞-動詞: 1
接尾辞-助詞: -
…

単語生起テーブル

名詞-東: 1
名詞-京都: 1
名詞-東: -1
動詞-住む: -1
接尾辞-都: -1
動詞-に: 1
…

- 学習の終盤は、**赤枠**と**黒枠**が重なる
- +1 -1 = 0 とコストが相殺されるので、更新が行われなくなる

未知語処理



- 辞書に無い単語の扱い
  - 字種をベースにしたヒューリスティックス
  - 例
    - カタカナの連続は1単語
    - 漢字は 1,2,3 文字の連続を1単語
    - アルファベットや数字はそれらの連続が1単語
  - 仮想的な辞書としてふるまう
- 未知語処理の振る舞いを再定義可能
  - 文字種の定義
  - 文字種ごとの振る舞いの定義

オープンソース形態素解析器

- 松本教授による prolog プロトタイプ
- 妙木,黒橋氏による C 実装
- コスト決定に2ヵ月

辞書の再編成

92年 0.6 94年 0.6 96年 3.1 03年 4.0 05年 5.1

JUMAN

パトリシアTRIE
back port

96.09.30
ChaSen

97年 1.0

- 浅原氏によるコスト推定法の研究
- 北内氏によるトライグラム拡張

03年 2.3.3

- 松本教授がNAISTに
- 統計処理によるコスト推定
- パトリシアTRIEによる高速化
- たつを氏が中心に開発

02.12.4
MeCab

ダブル配列
back port

06年 0.9

- ダブル配列による高速化
- 機能限定、よりシンプルに
- 開発言語を C++ に変更
- オブジェクト指向で書き直し

03年
Sen
MeCabの
Java port

06年 1.2.2

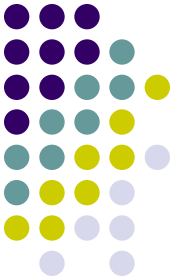
JUMAN/ChaSenとの違い

- JUMAN/ChaSen はあくまでも形態素解析器
  - 特定の品詞体系・辞書と密結合
  - 日本語に特化したハードコーディング
  - 日本語形態素に関する細かい振る舞いを扱える
- MeCab は単純マルコフ過程で記述できるタスクを汎用的に扱うためのフレームワーク
  - 入力・出力・辞書と疎結合
  - 日本語に特化していない
  - 汎用性が高い: iPhoneの仮名漢字変換

JUMAN/ChaSenとの違い

- 辞書とシステムの完全分離
  - 自然言語の複雑さはシステムではなく辞書/コストとして外部定義
  - システムは「ひらがな」「カタカナ」の区別すら知らない　(文字種の情報もすべて外部定義)
  - IPA dic, JUMAN dic, Unidic のサポート
  - JUMAN = 辞書 + システム

JUMAN/ChaSenとの違い



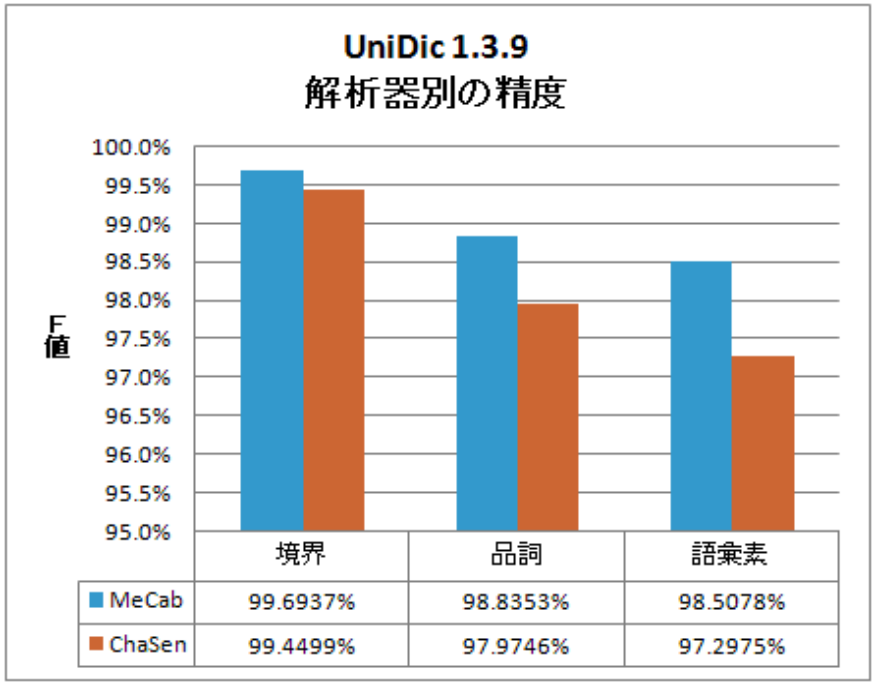
- 機能の選別
  - 前処理/後処理でできることはやらない
  - ChaSen の機能過多の反省
    - 改行処理, 連結品詞, 注釈, ChaSenサーバ.
  - 解析器にしかできない機能を提供
    - N-best 解, 制約つき解析, ソフト分かち書き
- 現代的な設計
  - 再入可能・スレッドセーフなライブラリ
  - クロスプラットフォーム (Linux/Windows/Mac)
  - C/C++/C#/Perl/Ruby/Python/Java .. ライブラリ
  - 安定動作

JUMAN/ChaSenとの違い

- 高速
  - ChaSen の2倍 Juman の 6倍以上高速
  - 事前にできることはすべてやっておく
  - 静的活用展開 vs 動的活用展開
- CRFによるコスト推定モジュールを同封
  - 今まではコスト推定はユーザ任せ
  - 辞書とタグつきコーパスがあればモデルを生成可能
  - Unidic プロジェクトへ貢献

JUMAN/ChaSenとの違い

- 高性能
  - CRF によるコスト推定
  - Unidic



  - JUMAN dic
    - MeCab-jumandic: 96.89 vs JUMAN 96.034

MeCab の辞書/設定ファイル

1. dic.csv (辞書定義)

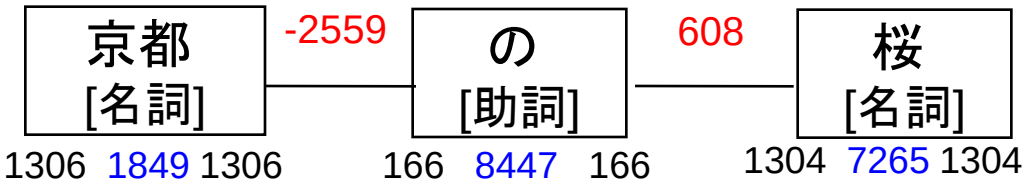
```
の,166,166,8487,助詞,格助詞,一般,*,*,*,の,ノ,
京都,1306,1306,1849,名詞,固有名詞,地域,一般,*,*,京都,キョウト,キョート
桜,1304,1304,7265,名詞,固有名詞,人名,名,*,*,桜,サクラ,サク
....
```

- 単語, 左文脈id, 右文脈id, 単語生起コスト, 素性列(CSV)
- 左文脈id,右文脈id を -1 にすると自動的に付与
- 素性は任意の情報(品詞,活用,読み等)をCSVで記述

2. matrix.def (連接コスト定義)

```
1306  166    -2559
1304 1303    401
166   1304    608
```

左文脈id 右文脈id 単語連接コスト



単語連接コスト
単語生起コスト

MeCab の辞書/設定ファイル

4. char.def (文字の定義)

```
NUMERIC        1 1 0
ALPHA           1 1 0
HIRAGANA       0 1 2
0x00C0..0x00FF ALPHA
0x3041..0x309F  HIRAGANA
...
```

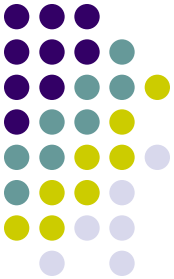
- 文字種ごとの動作
  未知語処理のタイミング
  文字種ごとにまとめるか
  何文字を単語とするか
-文字種の定義
  Unicode コードポイント

5. unk.def (未知語に付与する品詞の定義)

```
KANJI,1285,1285,11426,名詞,一般,*,*,*,*
NUMERIC,1295,1295,27386,名詞,数,*,*,*,*
ALPHA,1285,1285,13398,名詞,一般,*,*,*,*
```

-文字種を単語領域においた辞書ファイル

MeCab の素性フィールドの利用

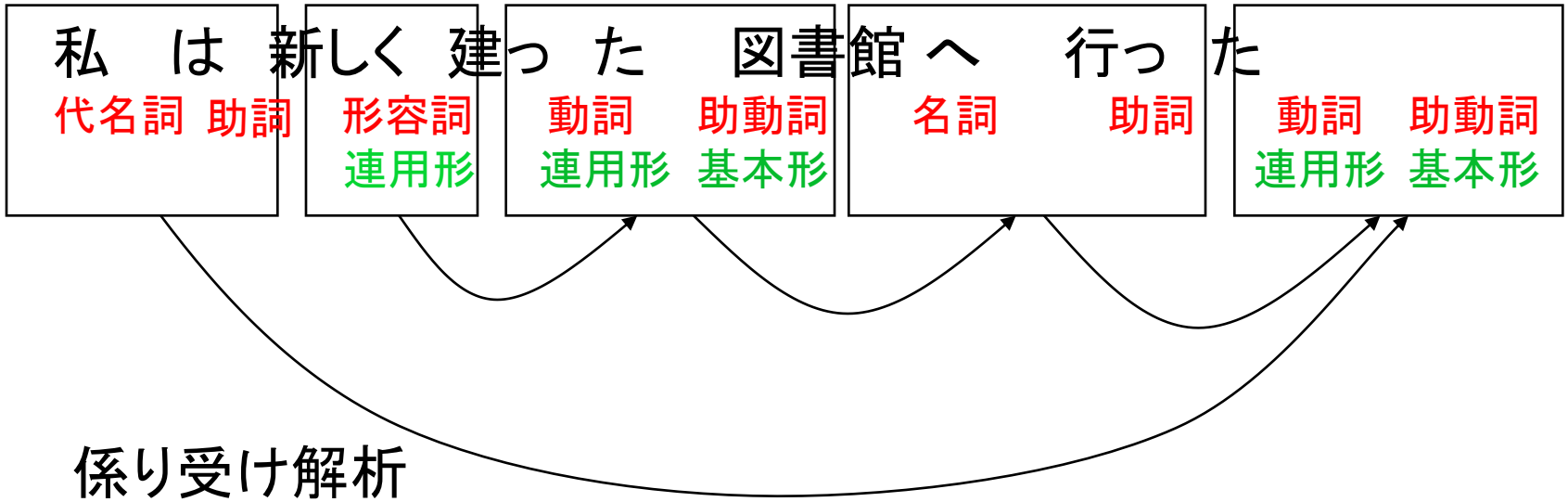


- 辞書の素性は CSV なら何でも可能
  - MeCab の動作は4カラム目まで決まる
  - 残りのCSVは単純にそのまま出力される
- 単語にさまざまな付加情報を付与可能
  - 意味情報
  - 英語の訳
  - URL
  - スパムスコア

```
の,166,166,8487,助詞,格助詞,一般,*,*,*,の,particle
桜,1304,1304,7265,名詞,固有名詞,人名,名,*,*,桜,サクラ,cherry
....
```

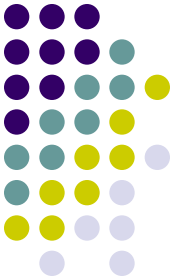
文節まとめ上げと係り受け解析

文節まとめ上げ



係り受け解析

日本語係り受け解析システム「CaboCha」



- Support Vector Machines (SVMs) を用いた学習に基づく係り受け解析器
- SVM の分類アルゴリズムの高速化手法である PKE を適用
- 上昇型で決定性の解析アルゴリズム (Cascaded Chunking Model)
- 解析中に得られた係り関係を素性として考慮する「動的素性」を利用

CaboChaによる文解析例

自民党は20日、衆院選小選挙区の第5次公認候補4人を発表した。

入力文

解析結果

```
        <ORGANIZATION>自民党</ORGANIZATION>は---------D
                          <DATE>２０日</DATE>、-------D
    <ORGANIZATION>衆院</ORGANIZATION>選小選挙区の-D   |
                                第５次公認候補-D |
                                        ４人を-D
                                       発表した。
EOS
* 0 5D 0/1 4.13561692
自民党      ジミントウ    自民党      名詞-固有名詞-組織                    B-ORGANIZATION
は          ハ           は          助詞-係助詞                  0
* 1 5D 2/2 2.98545849
２          ニ           ２          名詞-数                      B-DATE
０          ゼロ         ０          名詞-数                       I-DATE
日          ニチ         日          名詞-接尾-助数詞                        I-DATE
、          、           、          記号-読点                    0
* 2 3D 4/5 0.96527839
衆院        シュウイン   衆院        名詞-固有名詞-組織                          B-ORGANIZATION
選          セン         選          名詞-接尾-一般                    0
小          ショウ       小          接頭詞-名詞接続                      0
選挙        センキョ     選挙        名詞-サ変接続                  0
区          ク           区          名詞-接尾-地域                      0
の          ノ           の          助詞-連体化                0
* 3 4D 4/4 1.49313142
第          ダイ         第          接頭詞-数接続                  0
５          ゴ           ５          名詞-数                  0
次          ジ           次          名詞-接尾-助数詞                    0
公認        コウニン     公認        名詞-サ変接続                  0
候補        コウホ       候補        名詞-一般                  0
```

係り受け木

文節区切り

BIOタグによる
固有表現同定

MeCab/CaboChaに関する情報

- ホームページ:
  - MeCab:http://mecab.sourceforge.net/
  - CaboCha:http://chasen.org/~taku/software/cabocha/
- 茶筌の紹介と学習法についての資料:
  - 松本:形態素解析システム「茶筌」, 情報処理, Vol.41, No.11, pp.1208-1214, November 2000.
  - 浅原, 松本:形態素解析のための拡張統計モデル, 情報処理学会論文誌, Vol.43, No.3, pp.685-695, March 2002.
- MeCab のアルゴリズムについて
  - Taku Kudo, Kaoru Yamamoto, Yuji Matsumoto (2004)
    Appliying Conditional Random Fields to Japanese Morphological Analysis, EMNLP 2004
- CaboChaのアルゴリズムと高速化法についての資料
  - 工藤, 松本:チャンキングの段階適用による係り受け解析」, 情報処理学会論文誌, Vol 43, No. 6 pp. 1834-1842, June 2002.
  - 工藤, 松本:カーネル法を用いた言語解析における高速化手法 , 情報処理学会論文誌, Vol.45, No.9, pp.2177-2185, September 2004
- 茶筌と南瓜による文解析についての解説記事
  - 松本, 高岡, 浅原, 工藤:茶筌と南瓜による日本語解析--構文情報を用いた文の役割分類, 人工知能学会誌, Vol.19, No.3, pp.334-339, May 2004.