階層的モデルを用いた機械翻訳のためのフレーズアライメント

Graham Neubig[1,2,3], 渡辺 太郎[2], 隅田 英一郎[2],
森 信介[1], 河原 達也[1]

[1]京都大学 情報学研究科　　[2]情報通信研究機構 (NICT)
[3]日本学術振興会 特別研究員

1. はじめに

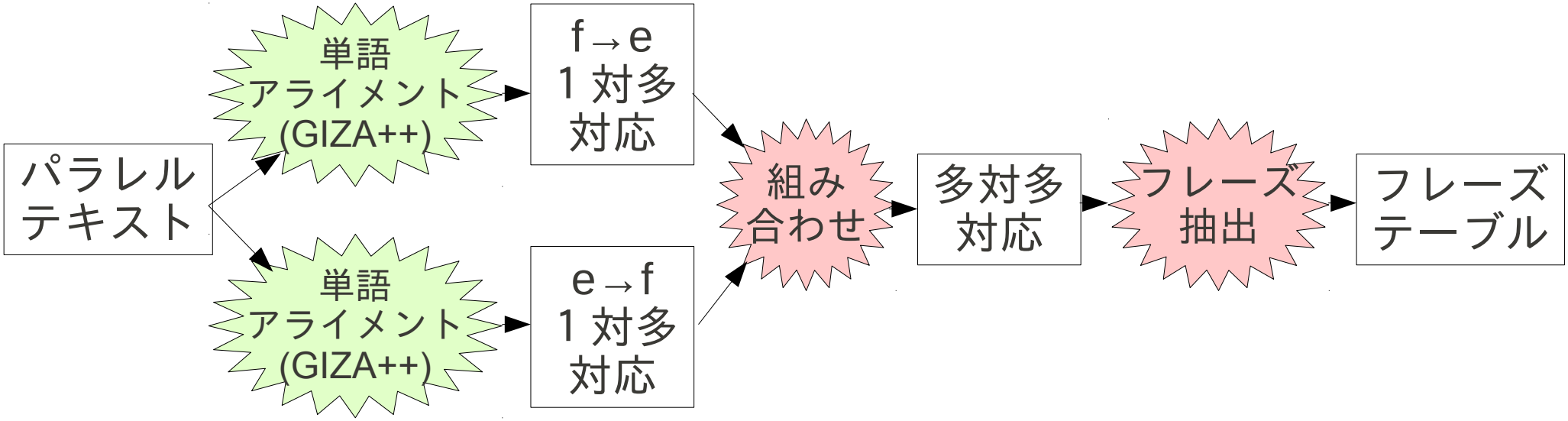
フレーズテーブル

- フレーズベース機械翻訳の中心
  - スコア付きの対訳フレーズ

原言語	目的言語	スコア
Mrs.	さん	0.05 0.20 0.005 1
Mrs. Smith	スミス さん	1.0 1.0 1e-05 1
Red	赤い	0.4 0.5 0.02 1
…		

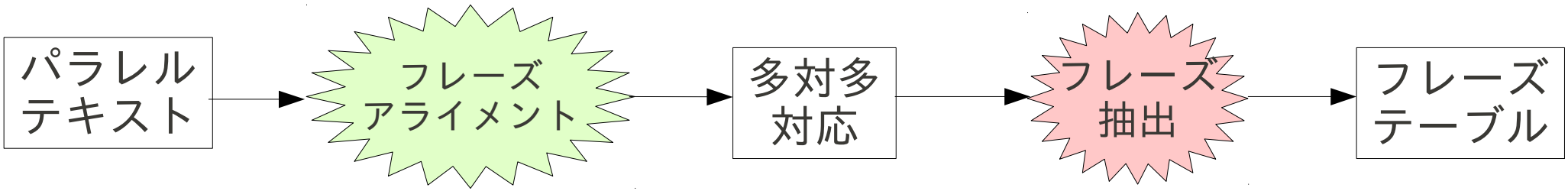
- 文単位でアライメントされた対訳コーパスから学習
  - フレーズのアライメントが必要

フレーズテーブル構築の従来法：1対多、組み合わせ、抽出



+ 意外と強力、 Moses などで広く使われる
- 複雑でヒューリスティックスがたくさん
- 段階に分かれるため、最終目的であるフレーズテーブルの構築に合わないアライメントを獲得
- 網羅的に抽出されたフレーズテーブルは膨大

フレーズテーブル構築の従来法：多対多、抽出



- 近年、直接多対多アライメントを獲得する手法が注目されている [Zhang+ 08, DeNero+ 08, Blunsom+ 10]

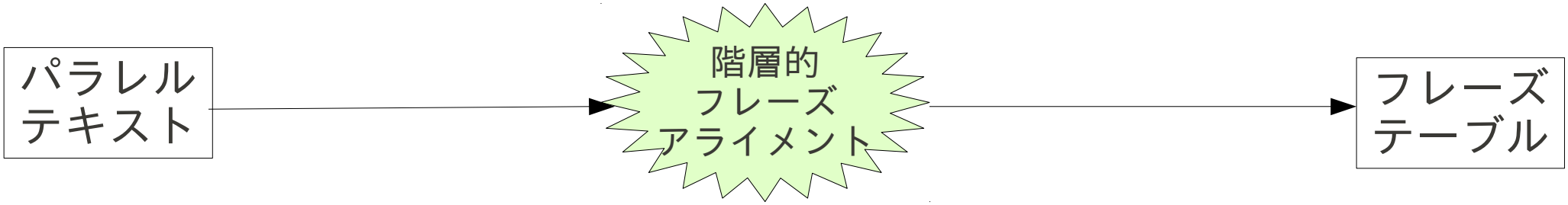
  \+ モデルが簡潔となり、精度も少し上がる

- 短いフレーズをアライメントし、フレーズ抽出で長いフレーズに組み合わせる

  \- まだ従来法の問題が多く残る

  - 膨大なフレーズテーブル、ヒューリスティックス、段階化により最適なアライメントが発見できない

提案手法：階層的モデル

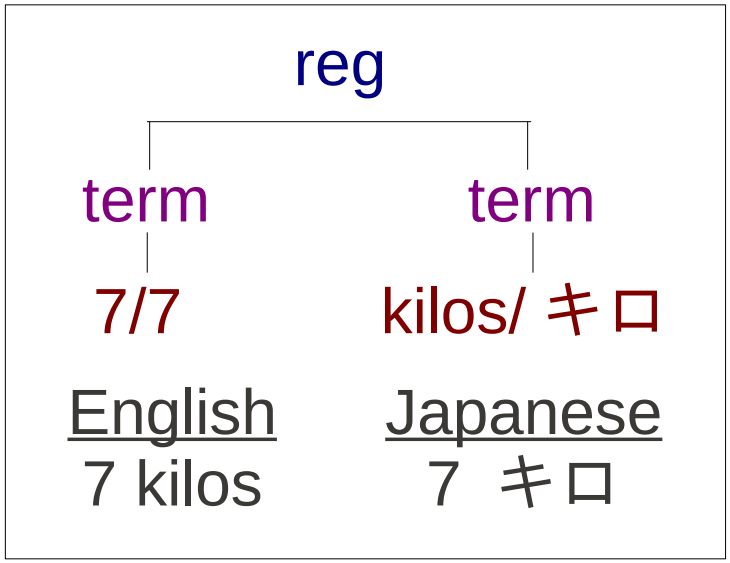


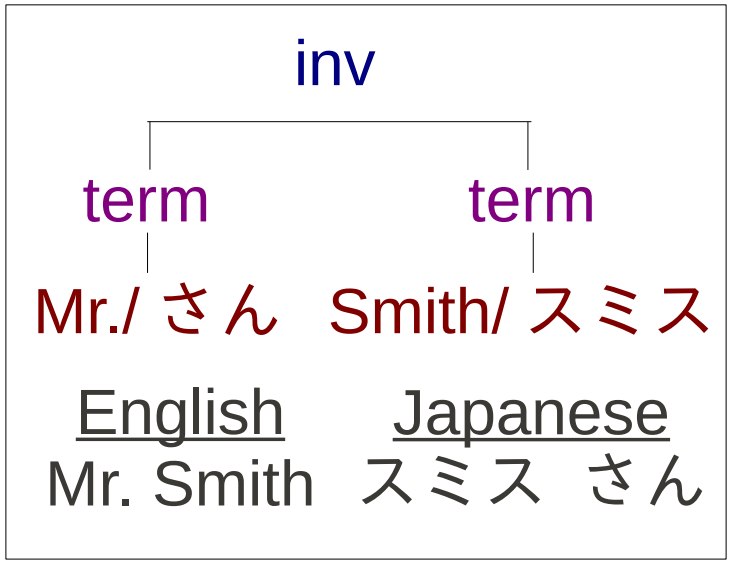
- 最小フレーズだけではなく、複数の粒度のフレーズをモデルに含む
  - + 直接フレーズテーブルをモデル化できる
  - + ヒューリスティックスの必要がなくなる
  - + 精度を保ちながらフレーズテーブルのサイズが減らせる
- Inversion Transduction Grammar (ITG) を用いた階層的モデルで実現

2. Inversion Transduction Grammar を用いたフレーズアライメント

Inversion Transduction Grammar (ITG)

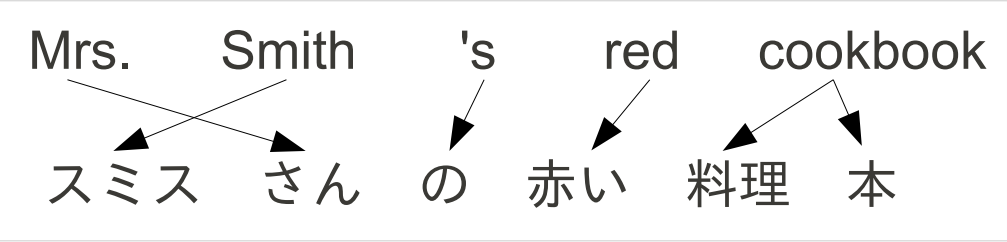
- 言語間をまたぐ文脈自由文法
  - 非終端記号は「普通 (reg)」と「倒置 (inv)」
  - 1 つの前終端記号「 term 」
  - 終端記号はフレーズペア

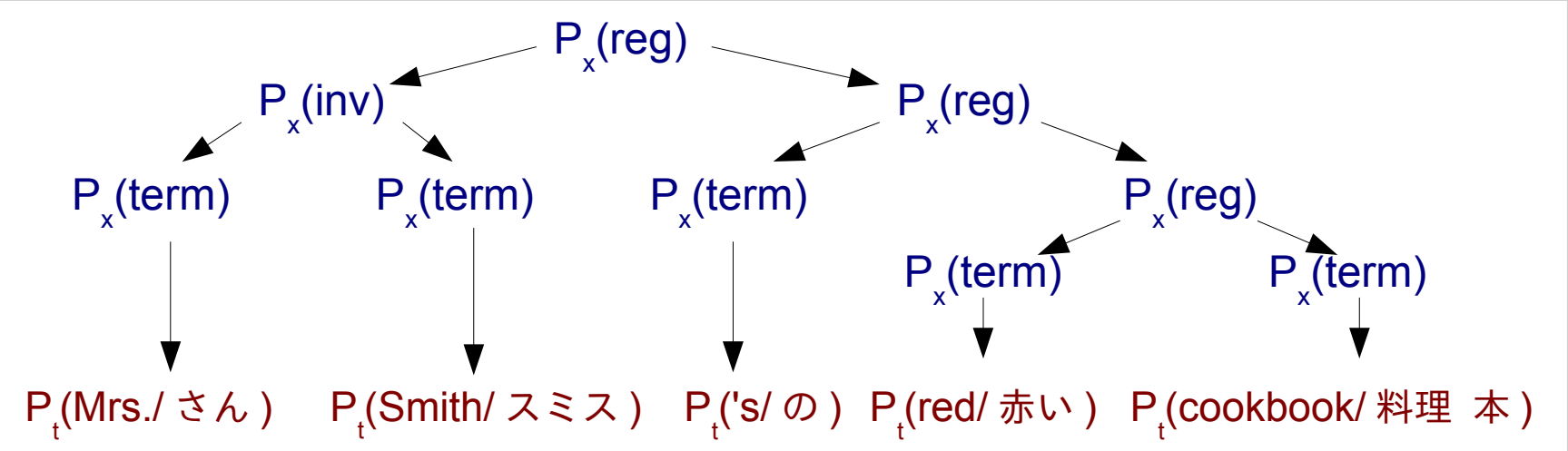




ITG を用いたパージング

- 非終端・前終端記号の記号分布 P_x, フレーズペア分布 P_t を利用し、両言語でパージングできる





- 最尤解も得られて、確率に従ったサンプリングも可能

フレーズペア・記号の頻度の求め方：ギブスサンプリング [Blunsom+ 10]

1. 各文のパーズを初期化し、頻度 c_x と c_t を数える
2. for 100 イタレーション
    1. for each 文 s in コーパス
        1. s の現在のパーズによる頻度を c_x と c_t から引く
        2. P_x と P_t の確率に基づいて s の新しいパーズをサンプリングする
        3. 新しいパーズによる頻度を c_x と c_t に足す
3. 得られた頻度に従って、フレーズテーブルを出力する

フレーズペア・記号確率の求め方

- フレーズペアと記号の頻度

c_t(Mrs./ さん)=12　c_t(Red/ 赤い)=3　c_t(Smith/ スミス)=0　…

c_x(reg)=415　c_x(inv)=43　c_x(term)=312

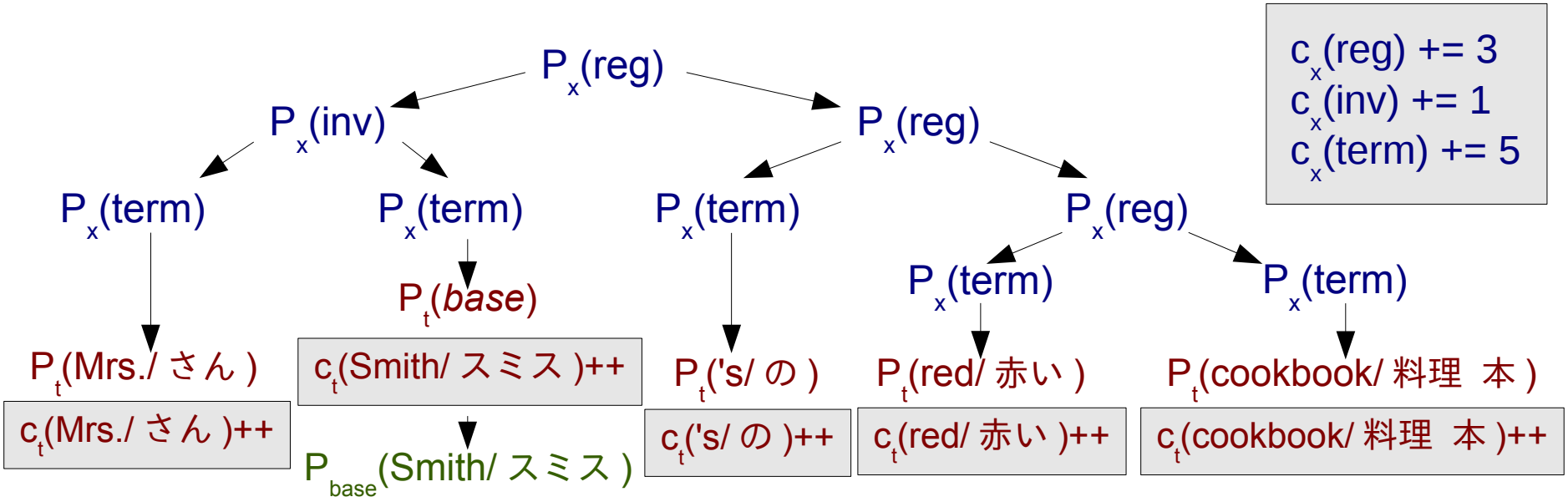
- スムージングを行う（例：加算スムージング）
  - 本研究のスムージング法は Pitman-Yor 過程（予稿を参照）

$$P_t(f,e)=\frac{c_t(f,e)+\alpha_t P_{base}(f,e)}{\sum_{f,e} c_t(f,e)+\alpha_t} \qquad P_x(x)=\frac{c_x(x)+\alpha_x/3}{\sum_x c_x(x)+\alpha_x}$$

- P_{base} は「未知フレーズペアモデル」
  - 全てのフレーズペアに少しの確率を与える
  - 先行研究で両方向の単語 Model 1 確率の相乗平均

従来モデルに追加される頻度

- 記号分布 P_x から生成されるものは c_x に加算
  フレーズペア分布 P_t から生成されるものは c_t に加算

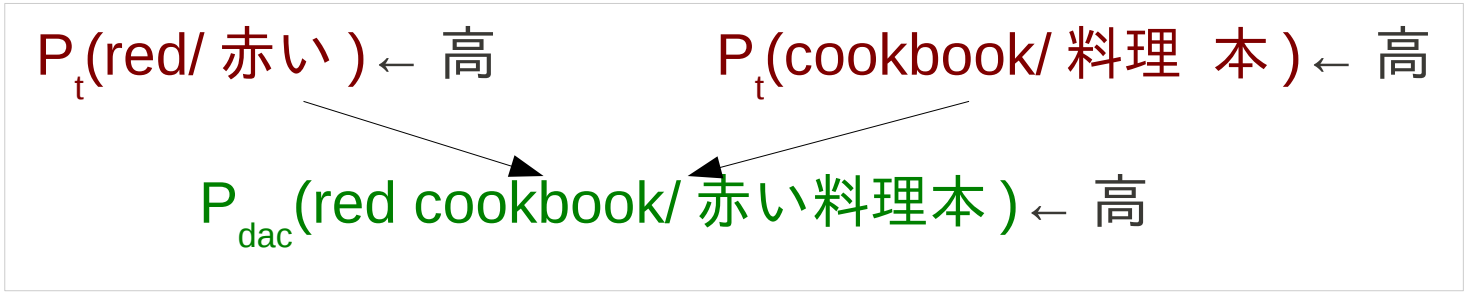


- 問題：最小フレーズのみが含まれる
  → アライメント後にヒューリスティックな抽出が必要

3. 提案手法：階層的なモデル

未知フレーズペアモデルを考え直す

- 未知のフレーズペアに与える確率
- 従来法：P_{base} は Model 1 確率の相乗平均
  - 高確率の単語ペアを含む→高確率のフレーズペア
- 提案手法：フレーズペアの組み合わせで未知のフレーズペアを構築するモデル P_{dac}



$$P_t(f,e) = \frac{c_t(f,e) + \alpha_t P_{dac}(f,e)}{\sum_{f,e} c_t(f,e) + \alpha_t}$$

P_{dac} の計算

- ITG と同じように、３通りのパターンで長いフレーズペアを短いフレーズペアから構築

普通： $P_x(reg)$ * P_t(red/ 赤い) * P_t(cookbook/ 料理 本) →
red cookbook/ 赤い 料理 本

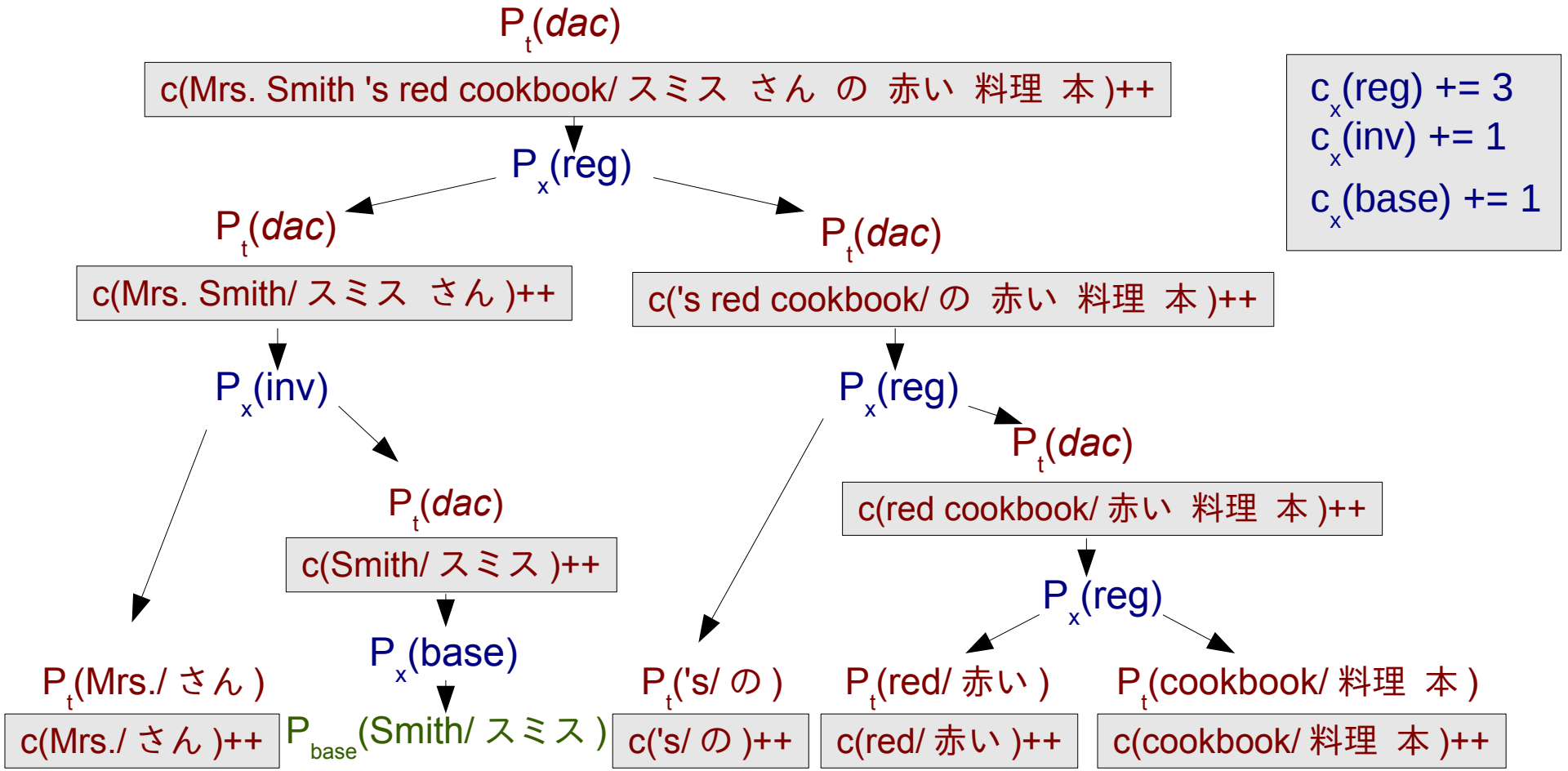
倒置： $P_x(inv)$ * P_t(Mrs./ さん) * P_t(Smith/ スミス) →
Mrs. Smith/ スミス さん

ベース： $P_x(base)$ * P_{base}(Smith/ スミス) →
Smith/ スミス

P_{dac} の計算に P_t を利用
P_t の計算に P_{dac} を利用
→再帰的なモデル！

提案手法の生成過程

- 複数の粒度のフレーズを分布 P_t から生成し、 c_t に加算



- 別途の抽出を行わなくても長いフレーズも獲得できる！

4. 実験評価

実験データ

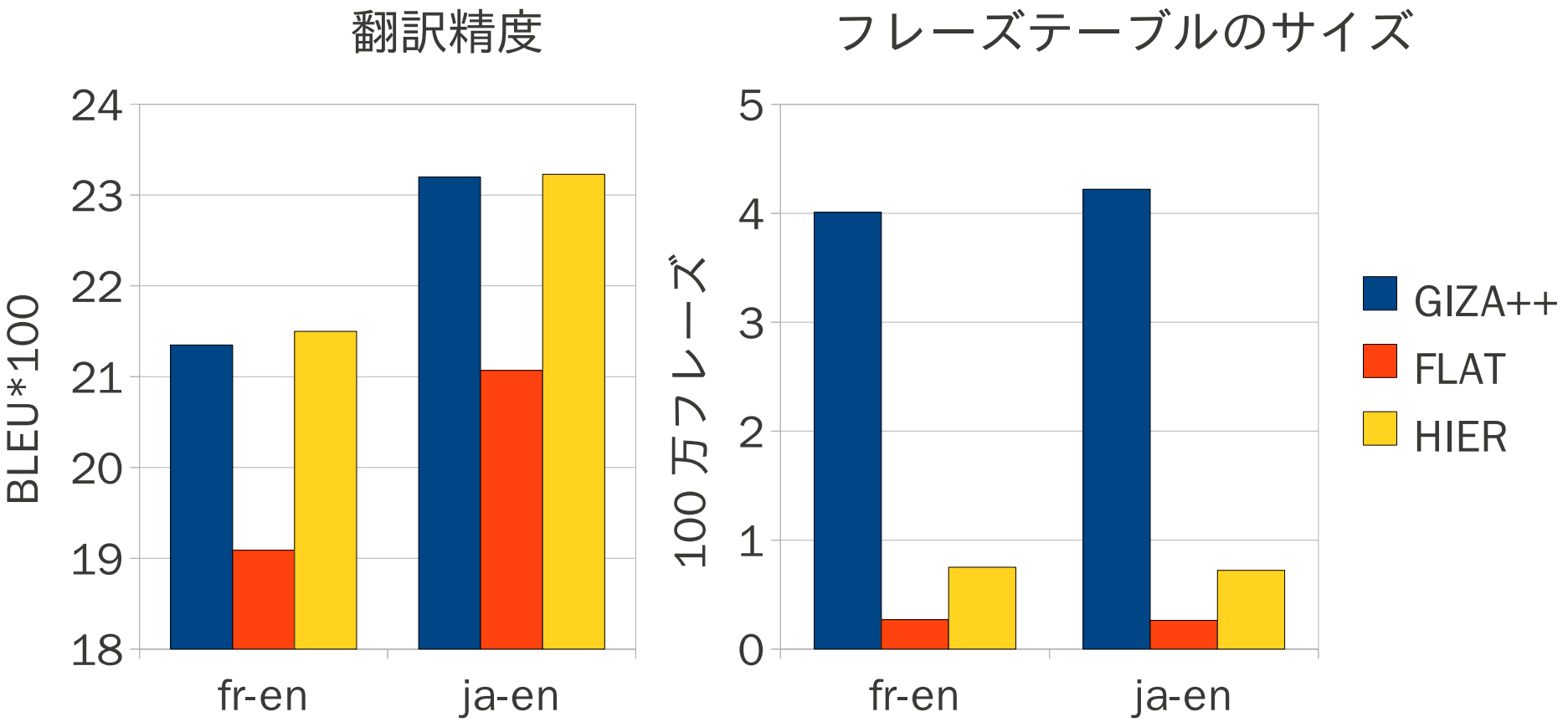
- 日英・仏英の 2 タスク
  - 日英： NTCIR の特許翻訳
  - 仏英： WMT10 の news-commentary タスク
- 学習データを小文字化、トークン化し、学習データを 40 単語以下の文とする

	WMT		NTCIR	
(単語数)	fr	en	ja	en
翻訳モデル	1.56M	1.35M	2.78M	2.38M
言語モデル	-	52.7M	-	44.7M
重み学習	55.4k	49.8k	80.4k	68.9k
テスト	72.6k	65.6k	48.7k	40.4k

実験設定

- デコーダとして Moses を利用
- BLEU を評価基準とする
- ３つのアライメント法：
  - GIZA++
  - 従来の ITG モデル (FLAT)
  - 提案手法の ITG モデル (HIER)
- ２つのフレーズ抽出法：
  - ヒューリスティックなフレーズ抽出
  - モデル確率 P_t を利用

実験結果



- GIZA++ はヒューリスティック、 FLAT と HIER はモデル確率
- GIZA++ と同程度の精度、フレーズテーブルは小さい
- 従来の ITG モデルに比べて高精度

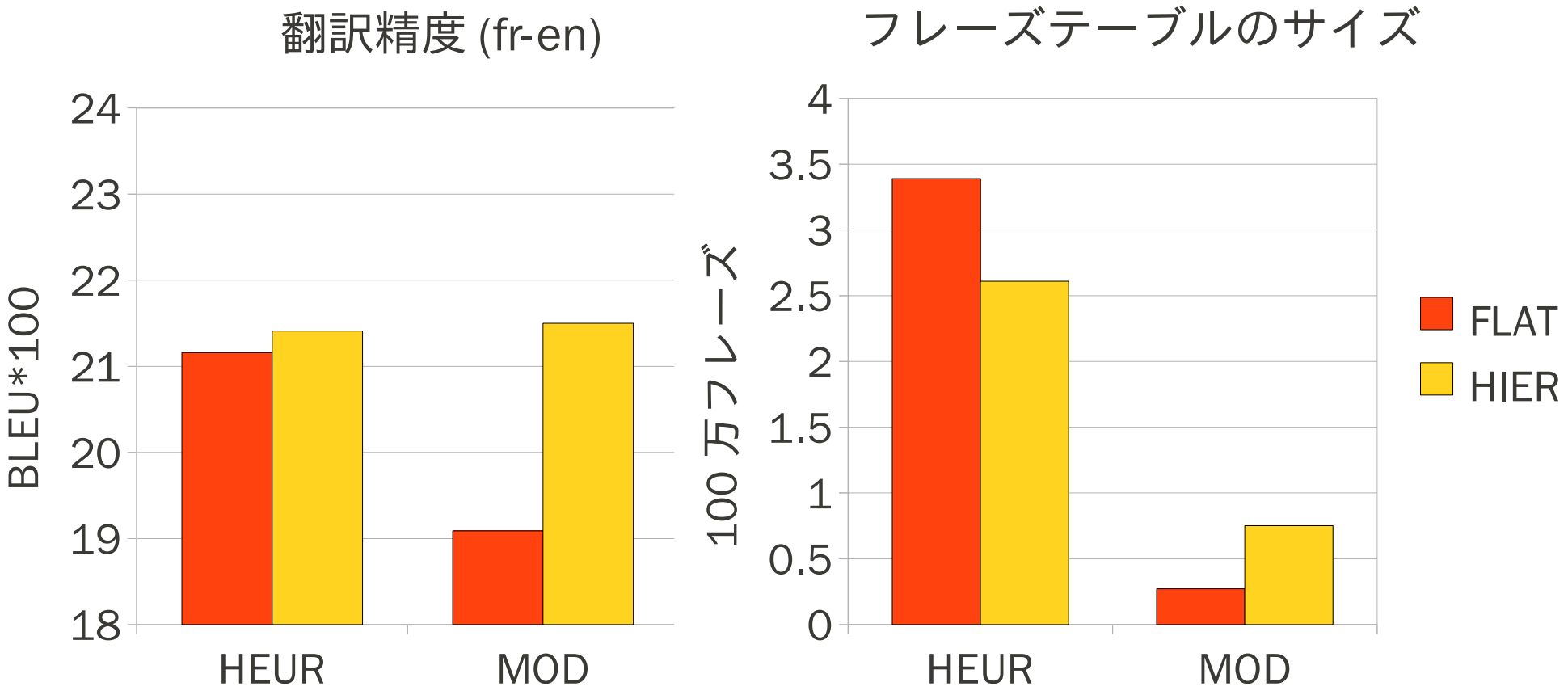
5. まとめ

まとめ

- 階層的モデルを利用することで、複数の粒度のフレーズをアライメントモデルに含める
- ヒューリスティックなフレーズ抽出を行わずに、簡潔なフレーズテーブルで高い翻訳精度を実現した
  - 完全な確率モデルが GIZA++/ ヒューリスティックスより高い精度になったのが初めて
- これからの課題：
  - 統語情報を利用した機械翻訳への適用
  - 分散処理による大規模データへの展開
- ソフトをオープンソースで公開する予定

ご清聴ありがとうございました

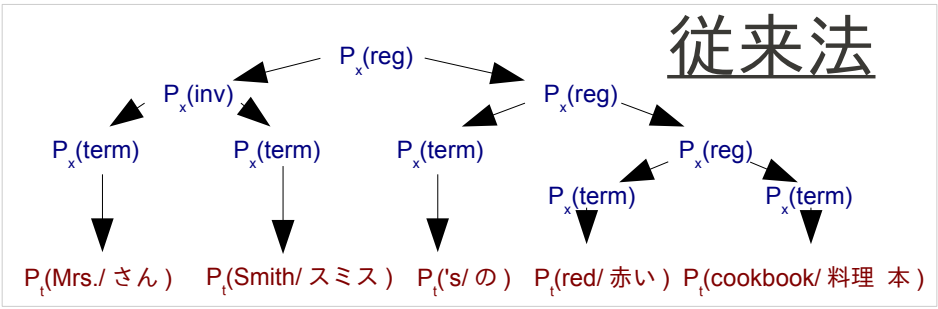
提案手法 vs. ITG アライメント＋ヒューリスティックなフレーズ抽出



- フレーズテーブルのサイズを増やすが、精度は HIER とモデル確率を利用した場合に劣る

提案手法の生成過程

- フレーズペアを生成してみる、バックオフで DAC



提案手法

$P_t(dac)$
P_x(reg)
$P_t(dac)$
$P_t(dac)$
P_x(inv)
P_x(reg)
$P_t(dac)$
P_x(base)
P_x(reg)
P_t(Mrs./ さん) P_{base}(Smith/ スミス) P_t('s/ の) P_t(red/ 赤い) P_t(cookbook/ 料理 本)

- 木の上から下まで、 P_t から生成