Innovative R&D by NTT

RIBES最新版の実装について
（最終？）

須藤 克仁
NTTコミュニケーション科学基礎研究所

この発表のねらい

- 公開版実装と論文の微妙なギャップを埋める
- 【公開版実装】RIBES-1.03.1 (2014/9/8)
- 【論文】
  - 平尾他「語順の相関に基づく機械翻訳の自動評価法」自然言語処理 21(3) pp.421-444, 2014
  - Isozaki et al., “Automatic Evaluation of Translation Quality for Distant Language Pairs,” Proc. EMNLP, pp.944-952, 2010

RIBES（ライビーズ）とは

Innovative R&D by NTT

- 「大域的な語順の差」に注目する自動評価尺度
  - 日英間の翻訳評価で人手評価との相関が高い
    - NTCIR-9 PatentMT Overview (pp.30-33)

- 最終的な定義式

語順相関の取れた単語数で計算

語順相関　単語適合率　簡潔ペナルティ

$$\mathrm{RIBES}(\mathcal{H},\mathcal{R}) = \frac{\sum_{h_i \in \mathcal{H}} \max_{r_j \in R_i} \{\mathrm{NKT}(h_i, r_j) \cdot P(h_i, r_j)^{\alpha} \cdot \mathrm{BP}_s(h_i, r_j)^{\beta}\}}{|\mathcal{H}|}$$

文スコアの算術平均

*平尾他(2014)より

文単位（複数参照訳）ではmax
コーパス単位では平均

NTT

単語対応の求め方 (1)

Innovative R&D by NTT

・同じ表層の単語が一回づつしか出現しない場合

We are pleased to inform you that your paper has been accepted .

We regret to inform you that your paper was not accepted .

同じ表層の単語が対応付いているものとする

単語単位でuniqueか？

NTT

単語対応の求め方 (2)

・同じ表層が複数出現する場合（EMNLP2010）

We are pleased ***to*** inform you that your paper submitted ***to*** the …

We regret ***to*** inform you that your paper submitted ***to*** AMTA …

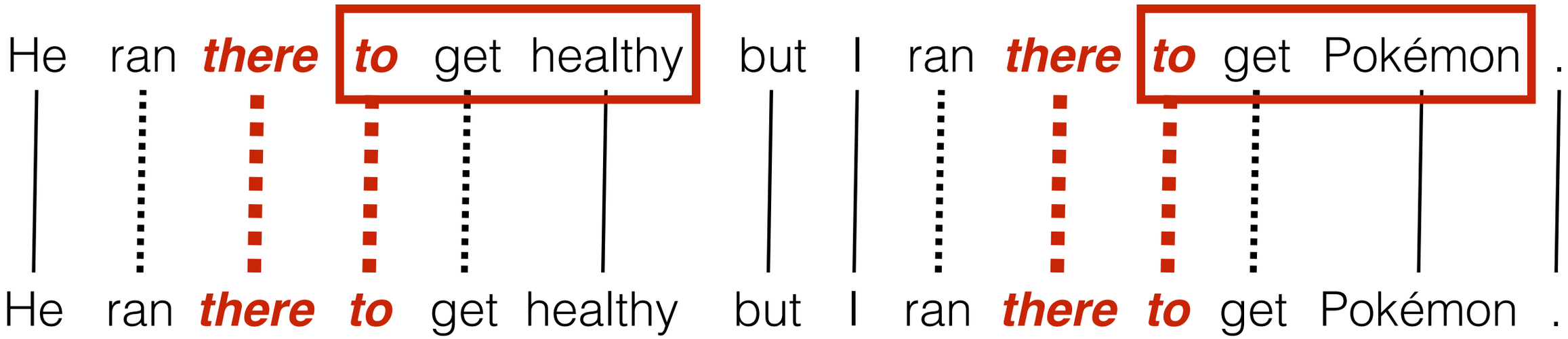
隣の単語を見て区別できるかどうか？

単語単位でuniqueか？ → 前/後の単語と組にしてuniqueか？

（+前1単語 → +後1単語 の順にチェック）

単語対応の求め方 (3)

・同じ表層が複数出現する場合（JNLP2014）



区別できるところまで文脈単語を見る

単語単位でuniqueか？ → 前/後の単語と組にしてuniqueか

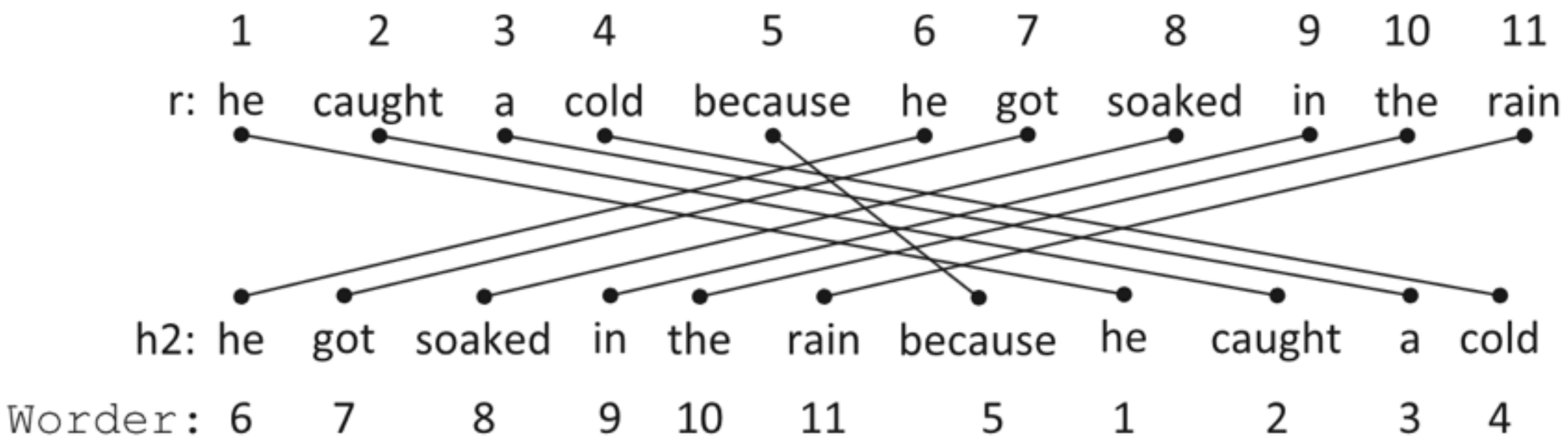
→ windowを広げながら前/後のn単語と組にしてuniqueか

（+前1単語 → +後1単語 → +前2単語 → +後2単語 → ...の順にチェック）

語順相関の求め方

Innovative R&D by NTT

・ 単語対応から「参照訳の対応単語ID」列を作る



*平尾他(2014)より

・ Worderに対して Normalized Kendall’s tau を計算する

・ τ = (#concordant - #discordant) / #pairs

・ NKT = (τ + 1) / 2

NTT

論文との微妙な違い

- Concordant pairsしか見ていない
  - 本来τ=(#concordant - #discordant) / #pairs
  - 実装では τ=2 x (#concordant / #pairs) -1
    - NKTで見れば #concordant / #pairs
- 対応付けアルゴリズムではtieが発生しないため (#concordant + #discordant = #pairs) 結果は同じ

論文にない実装上の機能

- Defaultではlowercasing （mtevalに合わせた）
  - Pythonの str.lower() 頼り
- 参照訳が空文の場合は計数しない（オプション）
  - リアルなデータの場合では起こり得るので…

小細工（たぶんあまり意味はないが…）

- 単語n-gramの出現数カウントの小細工
  - 各単語を Unicode char にマップする
    - 参照訳，評価対象訳に出てくる全単語を0始まりのIDにマップする（要はID付き辞書を作る）
    - 0x4e00からID分offsetした文字に置換する
  - 文字列上で「単語」n-gramを数える
    - str.count(s) は overlappingな計数をしない…
      （Google工藤さんの問題指摘後の調査で発覚）
    - str.find(s) でひとつずつ数えるハメに…
      （1.02.4での修正点）

おわりに

- 関係者の皆様に感謝申し上げます
  - 独自の実装を公開してくださっている皆様
  - Shared task評価に導入してくださっている皆様
- 残されている問題
  - 完全一致なのに1.0にならない
    - 単語対応の問題: スコア連続性の観点で未対応
- 何か問題などがあれば ribes@lab.ntt.co.jp まで

バージョン履歴

- 1.0 (2011/8/2) Initial release
- 1.01 (2011/8/10) Bugfix for NTCIR-9（空行エラー）
- 1.02 (2011/8/16) 拡張単語対応付け（小細工）の導入
  - 1.02.1 (2011/8/18) 文字変換バグの修正
  - 1.02.2 (2011/10/25) 標準出力の問題修正
  - 1.02.3 (2012/2/23) debug出力の問題修正
  - 1.02.4 (2013/12/17) 単語対応の問題修正
- 1.03 (2014/8/13) Python2.6対応, UTF-8化, 空白文字
  - 1.03.1 (2014/9/8) splitの互換性問題の修正