共参照解析を構造学習で解く際に
RNNでクラスタを覚えておく話

Learning Global Features for Coreference Resolution

Sam Wiseman, Alexander M. Rush, Stuart M. Shieber
Harvard University

NAACL2016

読み手: 三輪誠（豊田工業大学）

※一部の図は著者の論文・スライドから拝借

共参照解析

[]: 言及 (mention)

Dan Abrams: *... um and [I]$_1$ think that is what's - Go ahead [Linda]$_2$.*

Linda Walker: *Well and uh thanks goes to [you]$_3$ and to [the media]$_4$ to help [us]$_5$ ... So [our]$_6$ hat is off to all of [you]$_7$ as well.*

同じ実体 (entity) を指す言及のクラスタリング

記号の説明

$[I]_1$ $[Linda]_2$ $[you]_3$ $[the\ media]_4$ $[us]_5$ $[our]_6$ $[you]_7$

x_1 x_2 x_3 x_4 x_5 x_6 x_7

$y_1 = \epsilon$ $y_2 = \epsilon$ $y_3 = 1$ $y_4 = \epsilon$ $y_5 = \epsilon$ $y_6 = \epsilon$ $y_7 = \epsilon$

$z_1 = 1$ $z_2 = 2$ $z_3 = 1$ $z_4 = 3$ $z_5 = 4$ $z_6 = 4$ $z_7 = 5$

先行詞

クラスタ割当

x_n: 文書内出現順に並べた**言及**, $n \in \{1, \dots, N\}$

y_n: x_nの**先行詞** (の出現順) $\in \{1, \dots, n-1, \epsilon\}$

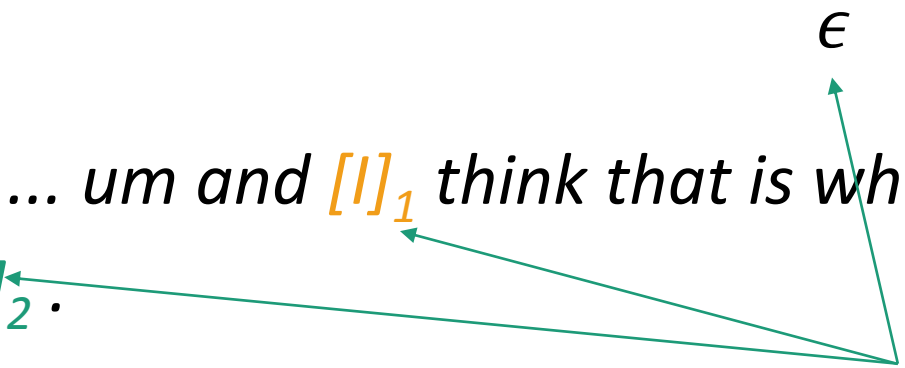
z_n: x_nの**クラスタ割当**

$X^{(1)}$	$X^{(2)}$	$X^{(3)}$	$X^{(4)}$	$X^{(5)}$
$[I]_1$ $[you]_3$	$[Linda]_2$	$[the\ media]_4$	$[us]_5$ $[our]_6$	$[you]_7$
$X_1^{(1)}$ $X_2^{(1)}$	$X_1^{(2)}$	$X_1^{(3)}$	$X_1^{(4)}$ $X_2^{(4)}$	$X_1^{(5)}$

$X^{(m)}$: 出現順の**クラスタ**, $m \in \{1, \dots, M\}$

$X_j^{(m)}$: クラスタ内で出現順に並べた言及

Mention-rankingモデル



Dan Abrams: *... um and* $[I]_1$ *think that is what's - Go ahead* $[Linda]_2$ *.*

Linda Walker: *Well and uh thanks goes to* $[you]_3$ *...*

文書の初めから言及の先行詞 (ϵ含む) を順に発見

- 簡単，効率的
- 先行詞を１つ見つければよいので，代名詞については見つけやすい

ニューラルネットによるMention-rankingモデル [Wiseman et al., 2015]

言及x_nと先行詞y_nのみから局所スコア関数を計算して個別に選択

$$y_n^* = \underset{y_n}{\text{argmax}}\, score(x_n, y_n) \triangleq \underset{y_n}{\text{argmax}}\, f(x_n, y_n)$$

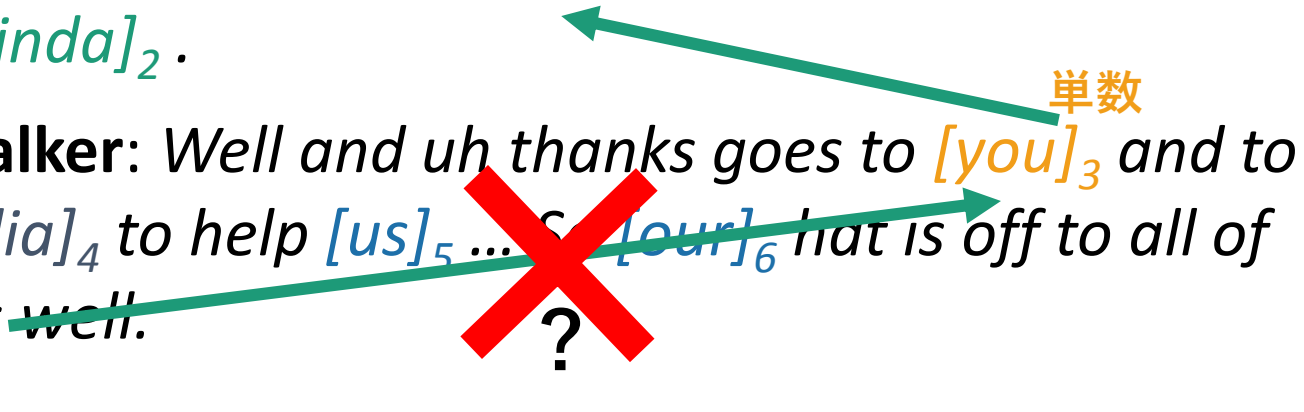
$$f(x_n, y_n) \triangleq \begin{cases} \boldsymbol{u}^T \begin{bmatrix} \boldsymbol{h}_a(x_n) \\ \boldsymbol{h}_p(x_n, y_n) \end{bmatrix} + u_0 \; if \; y_n \neq \epsilon \\ \boldsymbol{v}^T \boldsymbol{h}_a(x_n) + v_0 \qquad if \; y_n = \epsilon \end{cases}$$

- $\boldsymbol{h}_a(x_n)$: 言及x_nの素性$\boldsymbol{\phi}_a(x_n)$の埋め込み
  - $\boldsymbol{h}_a(x_n) \triangleq \tanh(\boldsymbol{W}_a \boldsymbol{\phi}_a(x_n) + \boldsymbol{b}_a)$
- $\boldsymbol{h}_p(x_n)$: 言及x_nと先行詞y_nの素性$\boldsymbol{\phi}_p(x_n, y_n)$ の埋め込み
  - $\boldsymbol{h}_p(x_n, y_n) \triangleq \tanh\left(\boldsymbol{W}_p \boldsymbol{\phi}_p(x_n, y_n) + \boldsymbol{b}_p\right)$
- $\boldsymbol{u}, \boldsymbol{v}, u_0, v_0, \boldsymbol{W}_a, \boldsymbol{W}_p, \boldsymbol{b}_a, \boldsymbol{b}_p$はパラメタ

大域素性 (global features)の必要性

Dan Abrams: *... um and [I]$_1$ think that is what's - Go ahead [Linda]$_2$.*

Linda Walker: *Well and uh thanks goes to [you]$_3$ and to [the media]$_4$ to help [us]$_5$ ... C[our]$_6$ hat is off to all of* ***[you]$_7$*** *as well.*



すでにわかっているクラスタの情報（=大域素性）が必要な場合がある（特に代名詞）

➔既に発見した共参照関係の情報を後ろで利用

大域素性を考慮した Mention-rankingモデル

$$\underset{y_1, \cdots, y_N}{\operatorname{argmax}} \sum_{n=1}^{N} score(x_n, y_n)$$

言及と先行詞と**直前までの**
クラスタリング結果から計算

局所スコア関数

大域スコア関数

$$\triangleq \underset{y_1, \cdots, y_N}{\operatorname{argmax}} \sum_{n=1}^{N} f(x_n, y_n) + g(x_n, y_n, \boldsymbol{z}_{1:n-1})$$

これまでの
クラスタ

※ 本研究では，言及の出現順に貪欲な探索をするので，目的関数を厳密には最大化しない

既存モデルにおける大域素性の問題

大域素性は入れても効いたり効かなかったり

- [Bjorkelund and Kuhn, 2014] ➔ 精度向上
- [Clark and Manning, 2015; Durrett and Klein 2014; Wiseman et al. 2015; Peng et al. 2015] ➔ 変わらない

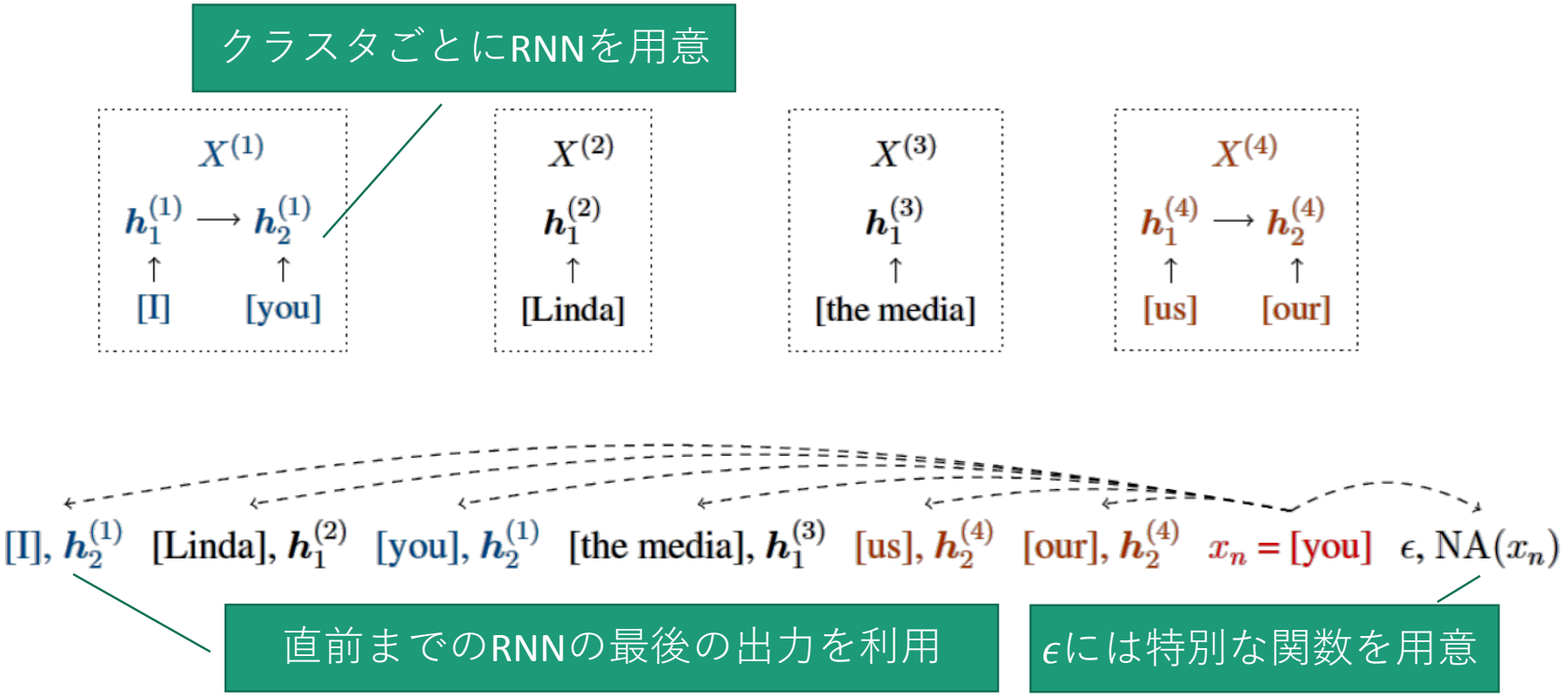
  ※ cluster ranking modelでは向上（次の発表）

なぜか？ ➔ 大域素性は定義しづらい

- 様々な大きさのクラスタから共通に比較できる，同じ大きさの素性を作るのが難しい
- 疎もしくは密になりやすい

提案手法

大域スコア$g(x_n, y_n, \boldsymbol{z}_{1:n-1})$をRNNを用いて計算



RNNを用いた大域スコア関数 - 先行詞がある場合

クラスタ内のRNNの最後の出力と対象の言及の内積

$$g(x_n, y_n, \boldsymbol{z}_{1:n-1}) \triangleq \boldsymbol{h}_c(x_n)^T \boldsymbol{h}_{<n}^{(z_y)}, if\ y_n \neq \epsilon$$

- $\boldsymbol{h}_c(x_n) \triangleq \tanh(\boldsymbol{W}_c \boldsymbol{\phi}_c(x_n) + \boldsymbol{b}_c)$
  - 言及x_nの素性$\boldsymbol{\phi}_c(x_n)$の埋め込み
    - $\boldsymbol{\phi}_c$は局所素性$\boldsymbol{\phi}_a$と同じ
- $\boldsymbol{h}_{<n}^{(m)}$は現在の単語以前でのクラスタ$m$内のRNNの最後の出力

$$\boldsymbol{h}_j^{(m)} = \mathbf{RNN}\left(\boldsymbol{h}_c\left(X_j^{(m)}\right), \boldsymbol{h}_{j-1}^{(m)}; \boldsymbol{\theta}\right)$$

  - クラスタmのRNNのj番目の出力
- $\boldsymbol{W}_c, \boldsymbol{b}_c, \boldsymbol{\theta}$はパラメタ

埋め込みもRNNも$\boldsymbol{h}$なので引数の有無で区別

RNNを用いた大域スコア関数 - 先行詞がない場合

言及の素性とすべてのRNNの最後の出力の平均

$$g(x_n, y_n, \boldsymbol{z}_{1:n-1}) \triangleq \boldsymbol{q}^T \tanh\left(\boldsymbol{W}_s \left[\begin{array}{c} \phi_c(x_n) \\ \sum_{m=1}^{M} h_{<n}^{(m)} \end{array}\right] + \boldsymbol{b}_s\right), if\ y_n \in \epsilon$$

- $\boldsymbol{q}, \boldsymbol{W}_s, \boldsymbol{b}_s$はパラメタ
- 「言及の埋め込みとϵに相当する埋め込みの積」と「RNNの出力の平均からの２階層NN」という解釈もできる

パラメタの学習

正解ラベル
（隠れ変数）

slack-rescalingとマージンを用いた目的関数

$$\underset{y_1,\ldots,y_N}{\operatorname{argmax}} \sum_{n=1}^{N} \Delta(x_n, y_n)\left(1 + score(x_n, y_n) - score\left(x_n, y_n^l\right)\right)$$

- y_n^lは現在のスコア関数が最大になるクラスタ内の言及
- 学習には正解のクラスタ $\boldsymbol{z}_{1:n-1}^{(o)}$ を利用 （予測は貪欲法)

$$score(x_n, y_n) = f(x_n, y_n) + g\left(x_n, y_n, \boldsymbol{z}_{1:n-1}^{(o)}\right)$$

予測クラスタ上での
学習やビーム探索は
効果なし

- slack-rescalingの値

$$\Delta(x_n, y_n) = \begin{cases} 0.5, \text{false link: 間違って前のクラスタに繋いだ場合} \\ 1.2, \text{false new:間違って新しいクラスタを作った場合} \\ 1.0, \text{wrong link: 繋ぐクラスタを間違えた場合} \end{cases}$$

利用した素性

Berkeleyの共参照解析システムを利用

約14,000種類

Mention Features (ϕ_a)
Mention Head, First, Last Words
Word Preceding, Following Mention
Words in Mention
Mention Synt. Ancestry
Mention Type
Mention Governor
Mention Sentence Index
Mention Entity Type
Mention Number, Gender, Person
Mention Animacy
Document Genre
Speaker
Mention contains Speaker
Normalized Document Position of Mention

約28,000種類

Pairwise Features (ϕ_p)
ϕ_a(Mention)
ϕ_a(Antecedent)
Mentions between Ment., Ante.
Sentences between Ment., Ante.
i-within-i
Same Speaker
Document Type
Ante., Ment. String Match
Ante. contains Ment.
Ment. contains Ante.
Ante. contains Ment. Head
Mention contains Ante. Head
Ante., Ment. Head Match
Ante. String Match with non-current Speaker

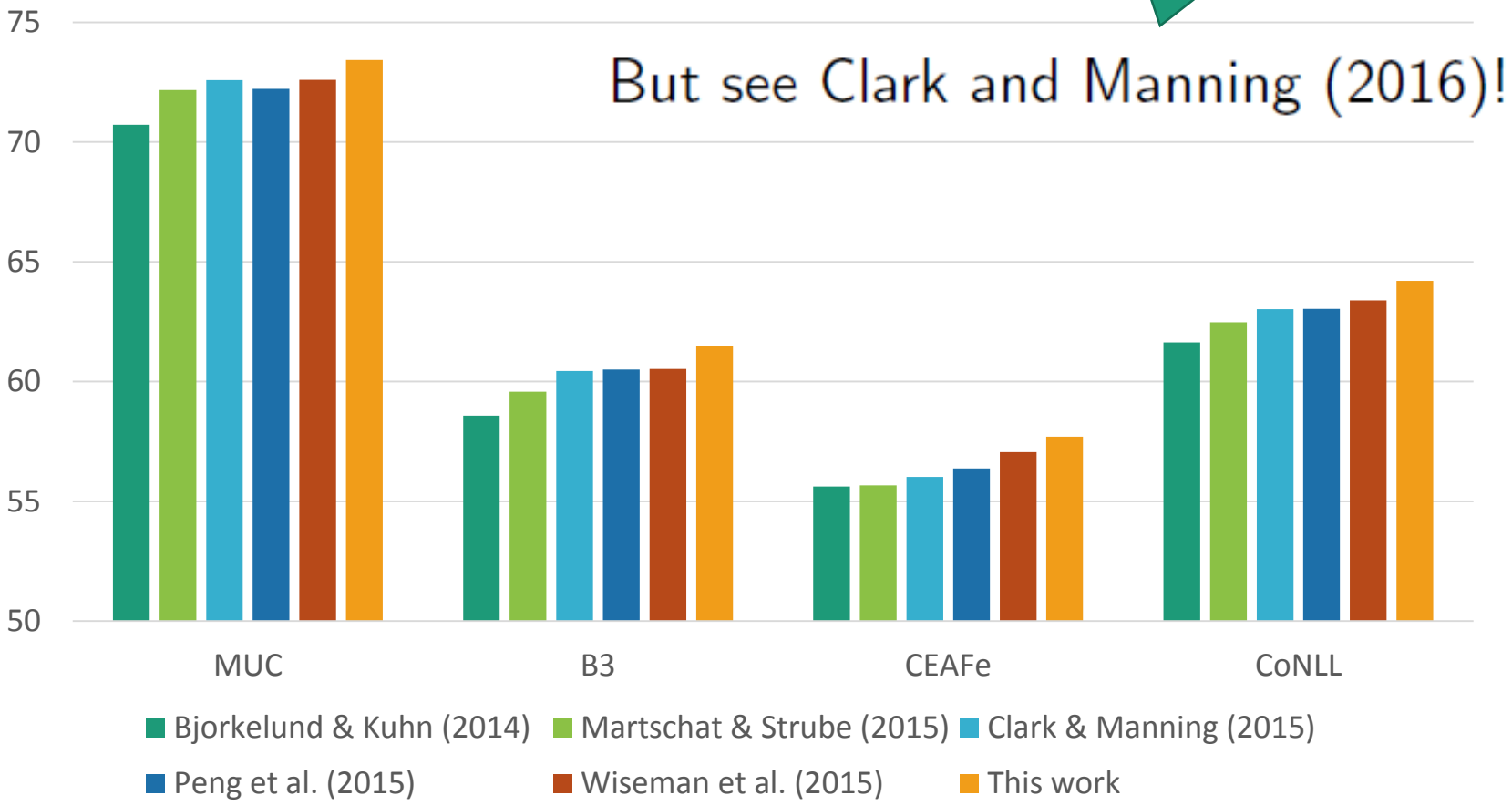
実験

- CoNLL 2012データセット
- 学習設定
  - 文書レベルでのミニバッチ
  - 照応詞かどうかの２値分類で事前学習
  - AdaGradを利用
    - 初期学習率を調節
    - クリッピング [-10, 10], Dropout (埋め込みに0.4, LSTMに0.3) の利用
  - RNNはピープホールなしの単層LSTM
  - 言及の埋め込み ($\boldsymbol{h}_a, \boldsymbol{h}_c$), クラスタ上のLSTM ($\boldsymbol{h}_m$) は200次元，言及と先行詞の埋め込み ($\boldsymbol{h}_p$) は700次元

    学習時は正解クラスタを使うので，クラスタ上のLSTMは事前計算可能
- 学習はGPU1台を用いて２時間程度

CoNLL2012での結果

著者のスライドより



開発データでの解析

- 予測時の直前までのクラスタ$\boldsymbol{z}$に予測を利用 vs. 正解クラスタを利用 (チートした場合の上限)
  - 65.47% vs 65.90% ➔ 上限より悪いが大きくは下がらない
- 局所スコア vs. 局所スコア+RNNによる大域スコア
  - 64.90% vs 65.47% (予測系列) ➔ 大域スコアは効果的
  - 代名詞のfalse linkの1,075から893 ➔冗言的代名詞を認識
  - headが一致する名詞のfalse linkも1,061から914
- クラスタのembeddingの平均 vs. RNN
  - 65.07% vs 65.90% (正解系列) ➔ RNNで順に見るのは大事

大域スコアの推移
(濃いほどスコアが高い)

"I had no idea I was getting in so deep," says Mr. Kaye, who founded Justin in 1982. Mr. Kaye had sold Capetronic Inc., a Taiwan electronics Maker, and retired, only to find he was bored. With Justin, he began selling toys and electronics made mostly in Hong Kong, beginning with Mickey Mouse radios. The company has grown -- to about 40 employees, from four initially, Mr. Kaye says. Justin has been profitable since 1986, adds the official, who shares **[his]** office... (nw/wsj/2418)

hisを見る時点で,
Justinが人でないと
わかっている

まとめ

- RNNを用いた大域素性を利用したモデル
  - うまくやれば大域素性は役に立つ
  - RNNは単純で効率的なクラスタの表現学習方法
  - 代名詞で特に大きな精度改善