【WWW 2014 勉強会】

Session 12: Content analysis 1 – entities

担当：欅惇志（東工大）

Session 12: Content analysis 1 - entities

- タスクの概要紹介
  - Named entity recognition (NER)
  - Named entity disambiguation (NED)
- 論文紹介
  - Discovering Emerging Entities with Ambiguous Names
    - Johannes Hoffart (Max Planck Istitute), Yasemin Altun (Google), Gerhard Weikum (Max Planck)
    - 新出 entity の情報を取得しつつ NED する
  - Effective Named Entity Recognition for Idiosyncratic Web Collections
    - Roman Prokofyev, Gianluca Demartini, Philippe Cudré-Mauroux (University of Fribourg)
    - ドメイン特化のテストコレクションで上手く NER する
  - Deduplicating a Places Database
    - Philip Bohannon, Nilesh Dalvi Marian Olteanu (Facebook), Manish Raghavan (UC Berkeley)
    - 地理データベースから重複データを除去する

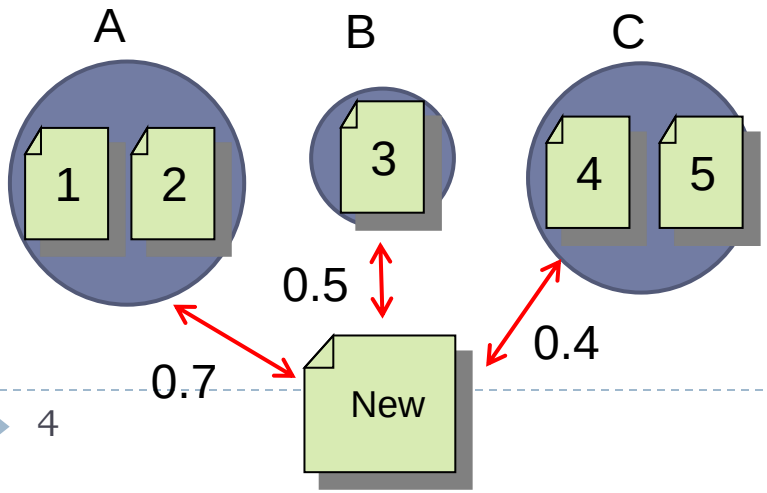
Named entity recognition

- テキスト中の固有表現 (named entity) を抽出
  - 自然言語処理の情報抽出タスクの一つ
    - テキスト分析の際に辞書として使うととても役立つ
  - 人名，地名，組織，事件 etc…
    - 属性の細かさ (people, politician, president) はアプリ次第
  - 代表的なアプローチ
    - ルールベース
      - 人手/bootstrap で獲得
      - 例) Mr. の後は people, male
        people の前によく出てくる語を people の新ルールとして追加
    - 機械学習ベース
      - 分類器作成
      - 例) 条件付き確率場: 教師データ -> 人手でラベル付与 (Obama -> people, politician, president), 素性 -> 周辺テキストから得られる各種統計量 (語，品詞 etc…)

Named entity disambiguation (event detection)

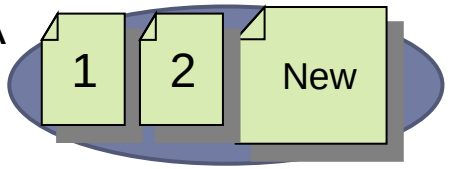
- NER で発見した entity (mention) が既存のどの entity を参照するのか，もしくは**新出 entity** であるのか判別
  - 新出がない場合は異なるクラスタリング手法が利用される
  - (乱暴にいうと) mention を含む文書と，既存 entity を含む文書群との比較

1. entity を含む文書 (New) と各クラスタの類似度を算出．各クラスの文書群の語との比較が多い (selected されたキーワードのみ使うこともよくある)



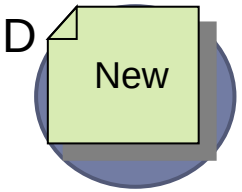
2. 最も高い類似度 > 閾値であれば，類似文書の属するクラスタへ追加

閾値
0.6



3. 最も高い類似度 < 閾値であれば，新しいクラスタ構成

閾値
0.8



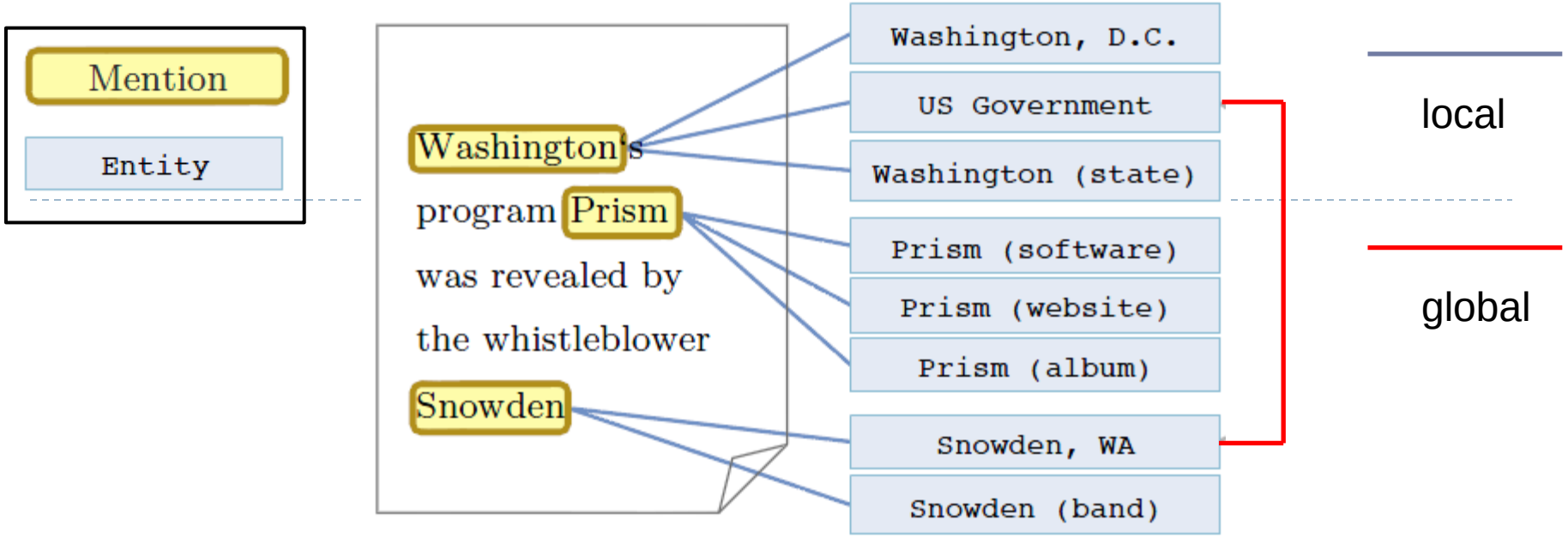
まとめ

- Named entity recognition
  - クラス判別
  - 文書のカテゴリ分類，クラスタリング，各種自然言語処理
  - 実用的な精度のツールも存在

- Named entity disambiguation
  - インスタンス判別
  - 知識ベース・知識グラフの作成

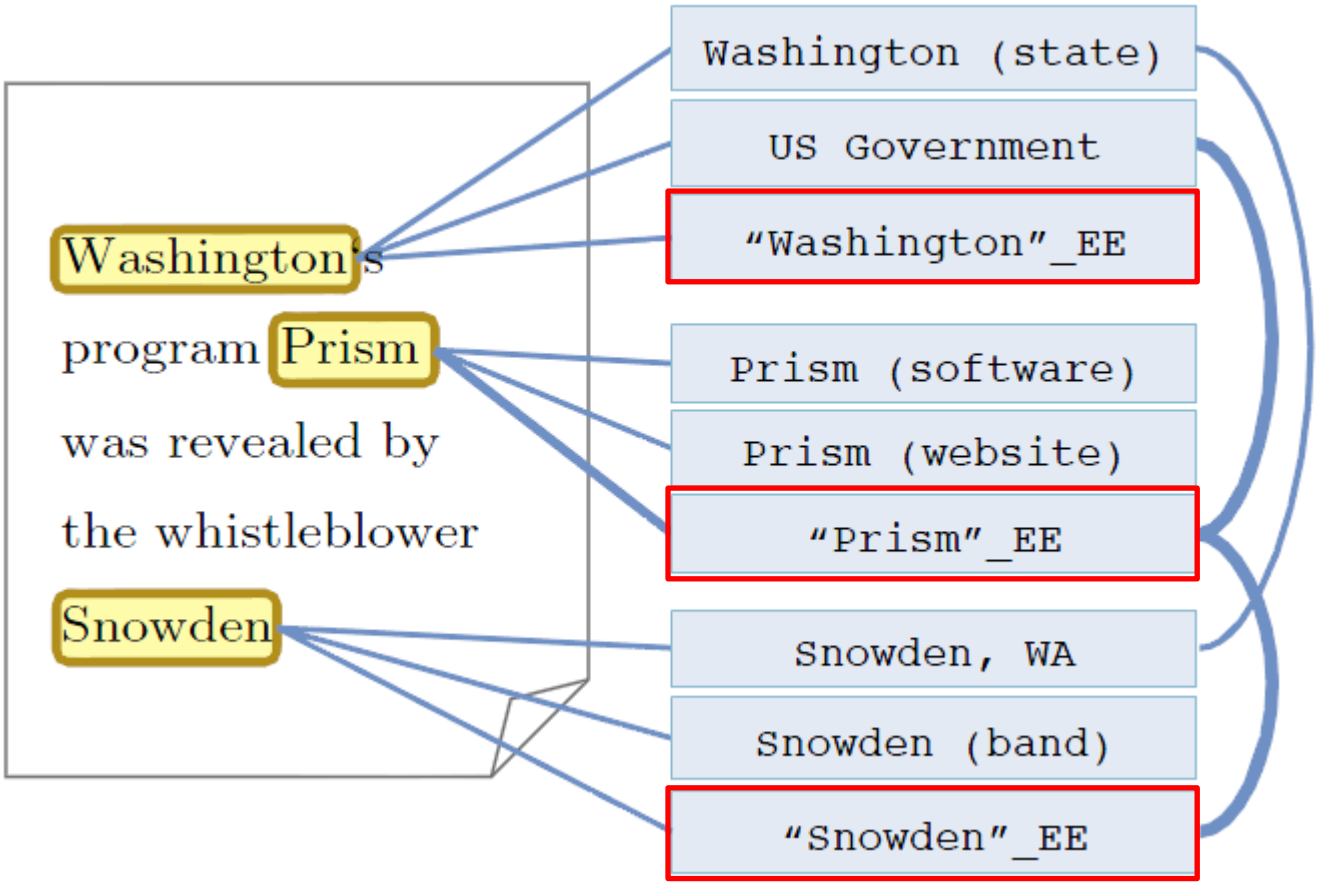
Discovering Emerging Entities with Ambiguous Names

- モチベーション
  - 実世界の entity は常に出現し続け，どんどん知識ベース (KB) に格納される
    - 新しい会社の設立
    - 新曲の発表
  - mention (入力文中の entity) の中には曖昧性を持つものあり
    - 過去のどれかの entity を参照 or 新出 entity を判別したい
      - 例) ハリケーン Sandy のニュースが出現したときに，既に KB に含まれる人名の Sandy と異なると判別したい
    - named entity recognition ではなく (既に entity だとは分かっている)，named entity disambiguation タスク



- 新出 entity 発見の常套手段
  - 対象 entity が KB の各 entity である可能性 (スコア) 算出
    - 閾値以上なら，最も高いスコアを持つ entity だと判別
    - 全ての entity のスコアが閾値以下であれば新出 entity
- スコア算出法 (文献 [23])
  - Local measure (冒頭で説明した event detection のようなもの)
    - 各 mention がどの entity であるのかの類似度に基づいて決定
    - 例) Washington -> Washington. D.C (0.3),
      Washington -> US Government (0.5)
  - Global measure
    - entity 間の類似度も考慮
    - 例) US Government は Snowden, WA と共起しやすい

提案



- 新出語 (EE) のときの entity も考える
  - 貢献 1
- スコア算出に使う情報足りないから外部から取ってくる
  - 貢献 2
- 評価実験の結果，最新の既存研究よりも提案手法は高精度

Effective Named Entity Recognition for Idiosyncratic Web Collections

- 概要
  - ニュース記事に対する NER は高精度
  - 学術論文など，特化したドメインでは上手く NER できない
    - novelty: 新出語がある
    - specificity: テキストの専門性が高い
  - 特化したドメインでも NER したい
    - 本研究では特化したドメインにおける entity を見つけること
    - general な文書中の entity は不正解
- 本研究における専門的文書集合の例
  - 学術論文
    - 学術リポジトリでは文書検索がされているが，entity 検索も有用

手法と結果

Feature name	Importance score
NN STARTS	0.3091
DBLP	0.1442
Component+DBLP	0.1125
Component	0.0798
VB ENDS	0.0386
NN ENDS	0.038
JJ STARTS	0.0364

- 手順
  - entity の候補 N-gram 作成
  - いろんな特徴量を使って学習 (決定木作成)
- N-gram を作成
  - 語 (の原型) の 2-gram を取り出す
  - 頻度が閾値以上の 2-gram から 3-gram を作成
    - ある 2-gram の後ろの語と，他の 2-gram の前の語が同じのとき
  - 5-gram まで繰り返す
    - 5-gram そのものは閾値未満でも entity の候補として取り出せる
    - recall 向上 (42.2% -> 96.1)
- いろんな特徴量
  - 品詞，句読点の近くかどうか，DBLP 中のキーワードが出現しているか，Wikipedia のタイトルか etc…
- 評価実験
  - 一番精度への影響が大きかったのは外部知識辞書関係の特徴量
  - baseline (最大エントロピー法，精度 69%) よりも高精度 (84%)

Deduplicating a Places Database

- Facebook の check-in データの多くは名称と位置のみ
  - ユーザが好き勝手名称を入力するので重複たくさん
  - 類似度の高いデータのペアは重複データとして除外
    - baseline (TF-IDF の内積) では上手く行かないので工夫必要
- 名称に関わる課題

$place_1$	$place_2$	
Fresca	*Fresca's Peruvian Restaurant*	duplicate
Newpark Mall Sears Outlet	*Newpark Mall Gap Outlet*	non-duplicate

  - 編集距離大きくても重複
  - 編集距離小さくても別
- 名称と位置に関わる課題
  - Bleeker street の二つの商店
    - "Bleeker Grocery and Convenientce" と "Bleeker delicatessen"
  - Bleeker は全体では低頻度語なので，TF-IDF 値が大きくなる
  - Bleeker street からの距離で，Bleeker の重み調整が必要
    - Bleeker street から離れた場所にある商店なら，二つの商店は重複データ

手法

- 語の重み付け
  - レストランやカフェなどの一般的な語より，Starbucks や subway などの特定性の高い語 (core) に高い重み
  - 左辺が最大となる B, C, z を求める (EM アルゴリズム)
    - $P(N|B,C,z) = \prod_{n\in N}\prod_{w\in n} C(w)^{z(w,n)} \cdot B(w)^{(1-z(w,n))}$

ある地名 n 中の w が core だと値が大きくなる　core でないと値が大きくなる

- ランドマークからの距離を考慮した語の重み付け
  - 地図を区画 (l) に分割し，区画ごとによく使われる名称はランドマークとして設定，重みを小さくする
  - 最尤法で確率モデルを求める
    - $B[l]_{ml}(w) = \frac{count(w;l)}{\sum_w count(w;l)}$
  - smoothing は上記の背景モデル (B(w)) で線形補間
    - $B[l](w) = \lambda B[l]_{ml}(w) + (1-\lambda)B(w)$

一緒に学習

- 結果
  - baseline (TF-IDF) より高精度，precision 及び recall は 90%