Learning Co-Substructures by Kernel Dependence Maximization

IJCAI'17

横井 祥[1,2], 持橋 大地[3], 高橋 諒[1], 岡崎 直観[4], 乾 健太郎[1,2]

[1] 東北大, [2]RIKEN/AIP, [3] 統数研, [4] 東工大

ERATO 感謝祭 SeasonIV

2017-08-04

導入：やりたいこと

① 獲得したい常識的知識

② コーパス内の生文
余計な語がたくさん入っている

"Bob **had dinner** with his friends at his favorite Japanese restaurant just now and he **is full**."

⟨**have dinner**, **be full**⟩

③ 知識に含める語を自動で選択したい

根付き部分木を教師なしで決定する問題だと思う

Table of Contents

関心：言語表現ペアの獲得・予測

関連する言語表現ペアの獲得・予測（NLP の中心課題のひとつ）

- コーパスから，関連する言語表現ペアを**収集**（**知識獲得**）
- 与えられた言語表現ペアに関連があるのかないのかを**予測**

たとえば

- 単語と単語の関係：概念同士の関係
  - 意味は近いか，上位下位関係を持つか
- 文と文の関係：命題同士の関係
  - 含意関係にあるか，因果関係にあるか
- **イベントとイベントの関係**　今回例として採用
  - 典型的に連続して生じるイベント対を獲得・予測したい [Schank&Abelson’77]
  - 例：⟨**have dinner, be full**⟩

問題：獲得パターンが固定

イベントペア獲得・予測の典型的プロセス [Chambers&Jurafsky’08]

1. **文ペアの収集：** 共参照項を持つ文対をコーパスから収集

   Tom killed nancy.

   co-referent

   The police

   arrested him immediately.

2. **予め決められた 抽象表現に変換：** 「述語動詞と登場人物の位置」に着目 ⟶ ⟨**X kill, arrest X**⟩
3. **モデル化：** PMI で関連の良さをモデル化

$$\mathrm{PMI}(X\,kill, arrest\,X) = \log \frac{p(X\,kill, arrest\,X)}{p(X\,kill)p(arrest\,X)}$$

一見問題なさそう

- **知識獲得**：可読的な知識 ⟨**X kill, arrest X**⟩
- **予測**：自己相互情報量（PMI）によるモデル化

問題：獲得パターンが固定

先の手法は **場合によって** 問題がある [Granroth-Wilding&Clark'16]

- ⟨“*Tom* had had absent repeatedly.”, “*He* was fired.”⟩
  → ⟨**X have**, ***fire*** **X**⟩
- ⟨“*Bob* has a talent for accounting work.”, “*He* was hired with favorable treatment.”⟩
  → ⟨**X have**, ***hire*** **X**⟩

この場合に 理想的な知識

- ⟨**X have *absent repeatedly*, fire X**⟩，⟨**X have *talent*, hire X**⟩

必要な情報はインスタンス毎に異なる

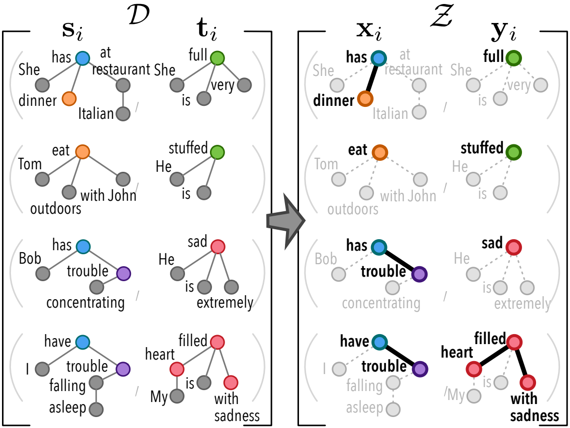
他，次のような語句の有無によってもイベントの意味が大きく変わり得る

- 否定表現
- 特定の条件を表す修飾節
- etc.

Table of Contents

解きたい問題

ペアをペアたらしめている部分構造を，**インスタンス毎に**，
教師なしで決定する（獲得する知識の “粒度” を教師なしで決定したい）

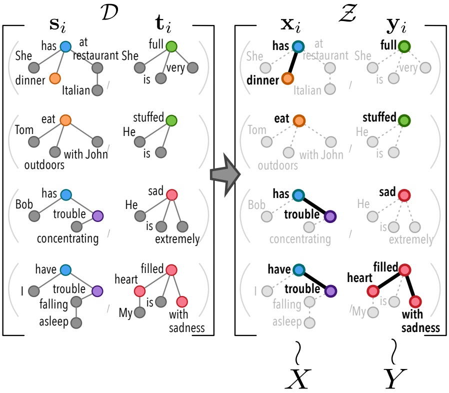


今回の設定：各文を依存構造木で表現

（根付き部分木の大きさを調整すれば表現の抽象度を調整できる）

定式化：従属性最大化

入力 文のペアの集合 $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^{n}$
出力 元の文の部分構造のペアの集合 $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$
目的関数 $\mathcal{Z} \overset{\text{i.i.d.}}{\sim} P_{XY}$ と見て maximize $D[P_{XY} \| P_X P_Y]$



cf. 特徴選択:入出力間の関連の良さを**従属性**に帰着 [Peng+'05][Song+'12]

従属性最大化に伴う問題

目的関数：データ・スパースネス

✘ 数千〜数百万オーダーの語彙 ⋯ の組合せ
$\longrightarrow$ 個々の部分構造は超低頻度
例：**have dinner at my favorite Italian restrant**
（ナイーブな最尤推定における）各 $p(\mathbf{x}, \mathbf{y}), p(\mathbf{x})$ は非常にスパース

$$\begin{aligned} I(X;Y) &= \mathrm{KL}[P_{XY} \| P_X P_Y] \\ &= \sum_{\mathbf{x}} \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}) p(\mathbf{y})} \end{aligned}$$

✘ 多様な言い換え表現（自然言語処理に常につきまとう問題）
例：**get angry** $\leftrightarrow$ **be offended**
$\longrightarrow$ 完全一致ではなく**類似度**に基づいて従属性を測りたい

探索：組合せ爆発
各 $\mathbf{x}_i$ 各 $\mathbf{y}_i$ についてそれぞれ部分木の取り方を考える必要がある

Table of Contents

Table of Contents

目的関数（従属性尺度）：HSIC

Hilbert–Schmidt Independence Criterion [Gretton+05]

- カーネル法ベースの独立性，従属性尺度

$$\mathrm{HSIC}(X, Y) = \mathrm{MMD}^2(P_{XY}, P_X P_Y)$$

- 出力 $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} P_{XY}$ に対する ***HSIC の推定量***

$$\mathrm{HSIC}(\mathcal{Z}; k, \ell) := \frac{1}{N^2}\mathrm{tr}(\mathbf{KHLH}) = \frac{1}{N^2}\mathrm{tr}(\tilde{\mathbf{K}}\tilde{\mathbf{L}})$$

$$\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j)) \in \mathbb{R}^{N\times N},\ \mathbf{L} = (\ell(\mathbf{y}_i, \mathbf{y}_j)) \in \mathbb{R}^{N\times N}$$

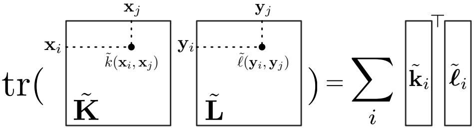
$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R},\ \ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}\ \ \text{(正定値カーネル)}$$

$$\tilde{\mathbf{K}} := \mathbf{HKH},\ \tilde{\mathbf{L}} := \mathbf{HLH}\ \ \text{(中心化グラム行列)}$$

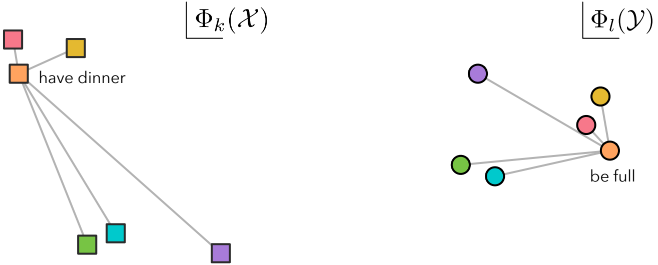
$$\mathbf{H} = (\delta_{ij} - N^{-1}) \in \mathbb{R}^{N\times N}$$

HSIC の推定量の気持ち

HSIC の推定量

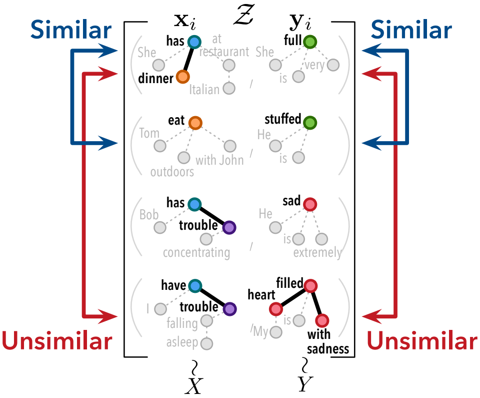


「HSIC の推定値が大きい」＝「カーネル関数（類似度）が定める距離が入った空間に放り込むと，前件側および後件側のフレーズの**相対的な位置関係がだいたい一致**する」



HSIC の推定量の気持ち

HSIC の推定値は以下の場合に大きくなる
- $\mathcal{X}$ 側が似ていれば $\mathcal{Y}$ 側も似ている
- $\mathcal{X}$ 側が似ていなければ $\mathcal{Y}$ 側も似ていない



✔ **類似性**に基づいた**一貫した**（従属性の高い）知識を期待できる

✔ 完全一致に基づいた数え上げではないのでデータ・スパースネスに対応できる

Table of Contents

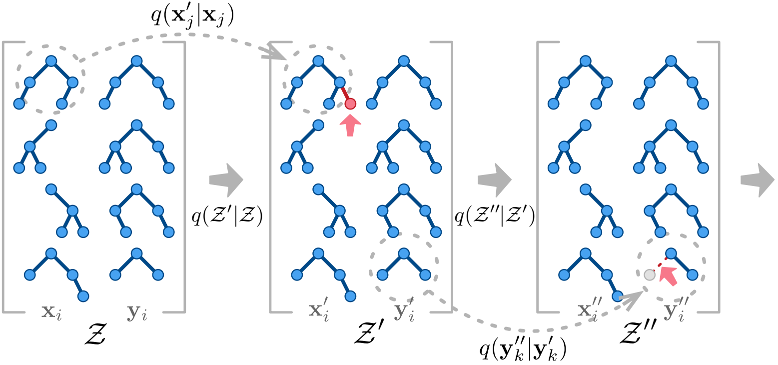
探索：Metropolis–Hastings

以下の分布上で Metropolis–Hastings (MCMC) でサンプリング

（焼きなましをしてもほとんど意味なし．適当な β = const. でほぼ一直線にサチる）

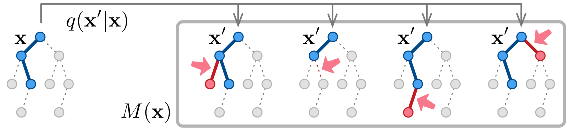
$$p(\mathcal{Z}; k, \ell, \beta) \propto \exp(\beta \cdot \mathrm{HSIC}(\mathcal{Z}; k, \ell))$$

少しずつ枝の刈り方を変えながら確率的に山登り



探索：提案分布

1. 現在の解候補：$\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$
2. **木を選択**：$\mathbf{x}_i$ または $\mathbf{y}_i$ をひとつ選択 $q(\mathbf{x}|\mathcal{Z}) = q(\mathbf{y}|\mathcal{Z}) = \frac{1}{2n}$
3. **枝を選択**：選択された $\mathbf{x}$ をわずかに変えて新しい部分構造 $\mathbf{x}'$ を作り ($q(\mathbf{x}'|\mathbf{x})$)，新しい解候補 $\mathcal{Z}' = \{\ldots, (\mathbf{x}'_i, \mathbf{y}_i), \ldots\}_{i=1}^n$ を得る



$$q(\mathbf{x}'|\mathbf{x}) = 1/|M(\mathbf{x})| \; (\mathbf{x}' \in M(\mathbf{x})),\;\; 0 \text{ (otherwise)}$$

4. 確率 $\min(1, r)$ で $\mathcal{Z}'$ を受理

$$r = \frac{p(\mathcal{Z}'; k, \ell, \beta)}{p(\mathcal{Z}; k, \ell, \beta)} \cdot \frac{q(\mathcal{Z}|\mathcal{Z}')}{q(\mathcal{Z}'|\mathcal{Z})}$$
$$= \exp(\beta \cdot (\mathrm{HSIC}(\mathcal{Z}'; k, \ell) - \mathrm{HSIC}(\mathcal{Z}; k, \ell))) \cdot \frac{q(\mathbf{x}|\mathbf{x}')}{q(\mathbf{x}'|\mathbf{x})}$$

5. 2–4 を繰り返し

計算コスト

- 中心化グラム行列 $\tilde{K}, \tilde{L}$ を構成するのは最初の 1 回だけ
  $O(N^2)$
- サンプリング毎にグラム行列 K, L を 1 行だけ更新
  $O(N)$
- $\rightarrow K, L$ を（ランク κ）不完全コレスキー分解してから $\mathrm{HSIC}(\mathcal{Z}; k, \ell)$ を計算
  $O(\kappa^2 N)$

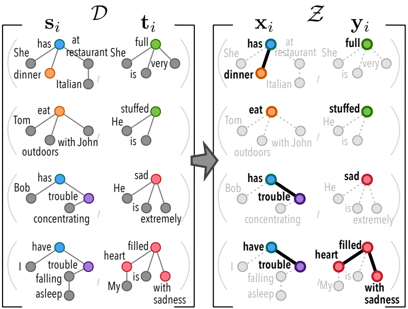
ここまでのまとめ

解きたい問題：ペアをペアたらしめている部分構造を教師なしで決定する

入力　文のペアの集合 $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^{n}$

出力　元の文の部分構造のペアの集合 $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$

目的関数　$\mathcal{Z} \overset{\text{i.i.d.}}{\sim} P_{XY}$ と見て max. $D[P_{XY} \| P_X P_Y]$（従属性最大化）



提案手法

- 目的関数：✘ データ・スパースネス＆多様な言い換え → ✔ HSIC
- 探索：✘ 組合せ爆発 → ✔ MH で確率的山登り

Table of Contents

Table of Contents

定性評価：小規模人工データからの知識獲得

入力：$\mathcal{D} = \{(\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^{12}$

$\mathbf{s}_i$	$\mathbf{t}_i$
I have had breakfast at my house .	I am full .
We had special dinner .	We are full .
I have had breakfast at ten .	I ’m full .
They had breakfast at the eatery .	They are full now .
She had breakfast with her friends .	She felt happy .
I had breakfast with my friends at my uncle ’s house .	I feel happy .
They had breakfast with their friends at the cafeteria .	They felt happy .
He had lunch with his friends at eleven .	He felt happy .
I had trouble associating with others .	I cry .
He had trouble with his homework .	He cries .
I have trouble concentrating .	I cry .
She had trouble reading books .	She cries .

類似度（カーネル）：

$$k(\mathbf{x}_i, \mathbf{x}_j) = \cos(\mathrm{ave}(\mathrm{wordvecs}(\mathbf{x}_i)), \mathrm{ave}(\mathrm{wordvecs}(\mathbf{x}_j)))$$

$$\ell(\mathbf{y}_i, \mathbf{y}_j) = \cos(\mathrm{ave}(\mathrm{wordvecs}(\mathbf{y}_i)), \mathrm{ave}(\mathrm{wordvecs}(\mathbf{y}_j)))$$

フレーズ間類似度の典型的な尺度．学習済み単語ベクトルを利用．

定性評価：小規模人工データからの知識獲得

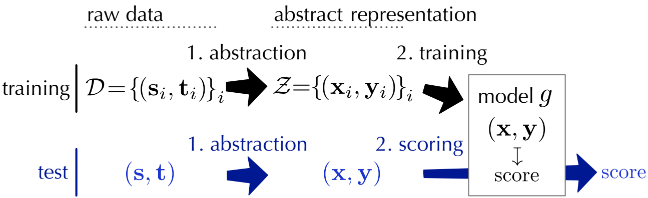
出力：$\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{12}$
太字：提案アルゴリズムが残した単語

$\mathbf{x}_i$	$\mathbf{y}_i$
I have ***had breakfast*** at my house .	I am ***full*** .
We ***had*** special ***dinner*** .	We are ***full*** .
I have ***had breakfast*** at ten .	I 'm ***full*** .
They ***had breakfast*** at the eatery .	They are ***full*** now .
She ***had breakfast*** with her ***friends*** .	She ***felt happy*** .
I ***had breakfast*** with my ***friends*** at my ***uncle*** 's ***house*** .	I ***feel happy*** .
They ***had breakfast*** with their ***friends*** at the cafeteria .	They ***felt happy*** .
He ***had lunch*** with his ***friends*** at eleven .	He ***felt happy*** .
I ***had trouble*** associating with others .	I ***cry*** .
He ***had trouble*** with his homework .	He ***cries*** .
I ***have trouble*** concentrating .	I ***cry*** .
She ***had trouble*** reading books .	She ***cries*** .

- ✔ 1度だけ出現する単語 (***dinner***, ***lunch***) が頻出語 (***breakfast***) との類似度に基づいて残される
- ✔ 第2ブロックの ***(with) friends*** が残され，左辺 → 右辺の予測を容易に

Table of Contents

定量評価：実コーパスを用いた関係予測



学習（知識獲得）

1. コーパスから共参照項を持つ文対を収集：$\mathcal{D}=\{(\mathbf{s}_i,\mathbf{t}_i)\}_{i=1}^n$
   例：⟨“*Tom* killed Nancy.”, “The police arrested *him* immediately.”⟩
2. 抽象表現に変換して保存：$\mathcal{Z}=\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^n$
   例：⟨**X kill, arrest X**⟩

予測

1. 文対 $(\mathbf{s},\mathbf{t})$ を抽象表現 $(\mathbf{x},\mathbf{y})$ に変換
2. 集めた $\mathcal{Z}$ を用い $(\mathbf{x},\mathbf{y})$ の関連の良さをスコアリング：$g(\mathbf{x},\mathbf{y};\mathcal{Z})$

評価尺度：AUC-ROC

定量評価：関連の強さの尺度

Poitwise Mutual Information [C&J'08]

$$\mathrm{PMI}(\mathbf{x}, \mathbf{y}; \mathcal{Z}) = \log \frac{N \cdot c(\mathbf{x}, \mathbf{y})}{c(\mathbf{x})c(\mathbf{y})}$$

Pointwise HSIC：中心化したカーネル密度推定

$$\mathrm{PHSIC}(\mathbf{x}, \mathbf{y}; \mathcal{Z}) := \frac{1}{N} \sum_{i=1}^{N} \tilde{k}(\mathbf{x}, \mathbf{x}_i) \tilde{\ell}(\mathbf{y}, \mathbf{y}_i)$$

$\tilde{k}(\cdot, \cdot)$ は既知のデータ点 $\{\mathbf{x}_i\}_{i=1}^{N}$ で中心化したカーネル

$$\tilde{k}(\mathbf{x}, \mathbf{x}') := k(\mathbf{x}, \mathbf{x}') - \frac{1}{N} \sum_{j=1}^{N} k(\mathbf{x}, \mathbf{x}_j) - \frac{1}{N} \sum_{i=1}^{N} k(\mathbf{x}_i, \mathbf{x}') + \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} k(\mathbf{x}_i, \mathbf{x}_j)$$

$\mathcal{X}$ に和が定義されていれば $\tilde{k}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \bar{\mathbf{x}}_i, \mathbf{x}' - \bar{\mathbf{x}}_i)$

PMI:MI ≈ PHSIC:HSIC

PHSIC は類似度でスムージングした PMI に見える

PMI で測る $(\mathbf{x}, \mathbf{y})$ の関連の良さ

- $\mathbf{x} = \mathbf{x}_i \wedge \mathbf{y} = \mathbf{y}_i$ なる $(\mathbf{x}_i, \mathbf{y}_i)$ が存在 $\longrightarrow$ PMI が**上昇**
- $\mathbf{x} = \mathbf{x}_i \veebar \mathbf{y} = \mathbf{y}_i$ なる $(\mathbf{x}_i, \mathbf{y}_i)$ が存在 $\longrightarrow$ PMI が**低下**

PHSIC で測る $(\mathbf{x}, \mathbf{y})$ の関連の良さ

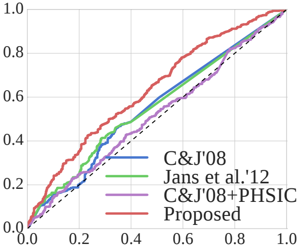
- $\tilde{k}(\mathbf{x}, \mathbf{x}_i)\tilde{\ell}(\mathbf{y}, \mathbf{y}_i) > 0$ なる $(\mathbf{x}_i, \mathbf{y}_i)$ が存在 $\longrightarrow$ PHSIC が**上昇**
  "$\mathbf{x} \approx \mathbf{x}_i \wedge \mathbf{y} \approx \mathbf{y}_i$" のとき上昇
- $\tilde{k}(\mathbf{x}, \mathbf{x}_i)\tilde{\ell}(\mathbf{y}, \mathbf{y}_i) < 0$ なる $(\mathbf{x}_i, \mathbf{y}_i)$ が存在 $\longrightarrow$ PHSIC が**低下**

PMI:MI ≈ PHSIC:HSIC

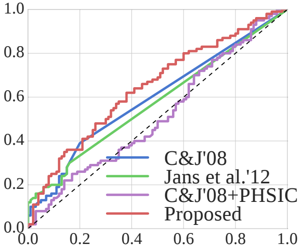
$$\mathrm{MI}(X, Y; \mathcal{Z}) = \frac{1}{N} \sum_i \mathrm{PMI}(\mathbf{x}_i, \mathbf{y}_i; \mathcal{Z})$$

$$\mathrm{HSIC}(X, Y; \mathcal{Z}) = \frac{1}{N} \sum_i \mathrm{PHSIC}(\mathbf{x}_i, \mathbf{y}_i; \mathcal{Z})$$

定量評価：実コーパスを用いた関係予測



(a) Gigaword $N = 16{,}748$



(b) Fairy Tale $N = 1{,}673$

Method	Abstraction	Model	Gigaword	Fairy Tale
[C&J'08]	Fixed (C&J)	PMI	0.553	0.596
[Jans+'12]	Fixed (C&J)	Conditional	0.556	0.576
[C&J'08] + PHSIC	Fixed (C&J)	PHSIC	0.518	0.518
Proposed	***Dynamic***	PHSIC	**0.633**	**0.646**

- ✔ 「インスタンス毎に注目すべき場所を決める」アプローチは予測精度にも寄与
- ✔ PHSIC という予測モデルでスコアが向上しているわけではない

Table of Contents

まとめ

問題：「ペアをペアたらしめている部分構造を探す」問題を提案

入力 文のペアの集合 $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^n$
出力 元の文の部分構造のペアの集合 $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$
目的関数 $\mathcal{Z} \overset{\text{i.i.d.}}{\sim} P_{XY}$ と見て max. $D[P_{XY} \| P_X P_Y]$（従属性最大化）

提案手法

- 目的関数：✘ データ・スパースネス＆多様な言い換え $\longrightarrow$ ✔ HSIC
- 探索：✘ 組合せ爆発 $\longrightarrow$ ✔ MH で確率的山登り

実験：イベントペアの獲得・予測

- 定量評価：提案手法が知識獲得の観点で理想的に動く
- 定量評価：インスタンス毎の抽象化が予測精度に貢献

今後の取り組み

- 高速化：現状数万オーダー $\longrightarrow$ 数百万オーダー
- より精緻な類似度関数の導入：構造カーネル
- 他タスクへの適用