従属性に基づく事態間関係知識の粒度調整

横井 祥 † 持橋 大地 ‡ 岡崎 直観 † 乾 健太郎 †
† 東北大学大学院 情報科学研究科 ‡ 統計数理研究所
{yokoi,okazaki,inui}@ecei.tohoku.ac.jp daichi@ism.ac.jp

1 はじめに

〈店に入る，注文をする〉のようによく共起する事態（イベント）の対を表す**事態間関係知識**は，計算機による自然言語理解の重要なコンポーネントであり，コーパスからこれを自動で獲得する研究がこれまで多く行われてきた [3, 1, 5]．事態間関係知識獲得の典型的なプロセスを，Chambers ら [1] の手法を例に挙げて以下に述べる．

(1) **コーパスからの知識獲得**：はじめに事態の形式・表現が決められる．Chambers らは事態の形式を「述語動詞＋注目しているエンティティの位置」としており，コーパスから得られる事態間関係知識は〈X kill, arrest X〉や〈purchase X, acquire X〉といった形式になる．具体的な知識獲得の手順としては，まずコーパスから共参照項を持つ文対を集め（例：〈‘*Tom* killed Nancy.’, ‘The police arrested *him* immediately.’〉），ここから述語動詞と共参照項のみを取り出し知識として獲得する（例：〈X kill, arrest X〉）．知識には述語動詞と共参照項**以外**の語，たとえば ‘police’ や ‘immediately’ は含まれない．他の先行研究でも，**知識の形式があらかじめ決められ**，これに合わせてコーパスから知識を獲得するという点は同様である [3, 5]．

(2) **獲得された知識に対する信頼度のスコアリング**：Chambers らの場合は，PMI を用いて事態間関係知識のスコア（事態同士の関連の良さ）を計算した．つまり〈X kill, arrest X〉に対して次のスコアが付与される：

$$\mathrm{pmi}(\text{‘X kill’}, \text{‘arrest X’}) = \frac{p(\text{‘X kill’}, \text{‘arrest X’})}{p(\text{‘X kill’})p(\text{‘arrest X’})}.$$

粒度調整の必要性とスパースネス問題

しかし，上述の Chambers らの手法に代表されるような獲得する知識の候補の形を予め固定的に限定するアプローチには問題がある．例えば、事態間関係の表現を〈X kill, arrest, X〉のように述語（kill や arrest）と共参照項（X）の組に限定してしまうと，〈X have absence, fire X〉や〈X have opportunity, hire X〉のような知識を獲得することができない．本稿ではこのように，どこまで周辺文脈を知識に入れるかを選択する問題を「**粒度調整の問題**」と呼ぶ．粒度をうまく調整しながら知識を獲得するには，様々な粒度の候補で知識を獲得し，その中から統計的に良い候補を何らかの方法で選択する必要がある．一方で，周辺文脈を知識に入れれば入れるほど個々の事態表現の生起は**スパース**になり，PMI をはじめとした統計的なスコアリングが困難になる．

これに対し本研究では，知識の粒度を教師なしで調整しつつ，かつスパースネスの問題にも対処するための手法を示す．

粒度調整問題の定式化

はじめに，知識の粒度を教師なしで調整する問題を定式化する．事態間関係知識獲得は次の特徴を持つ：(i) 粒度調整を行う前の文のペアは簡単に収集することができる．たとえば，ディスコースマーカー ‘because’ で結ばれた 2 文の集合，共参照項を持つ連続した 2 文の集合をコーパスから容易に収集できる．(ii) しかし適切な粒度，すなわちこれらの文の中で知識に含めるべき語がどれであるかは非自明である．したがって本稿では，文のペアの集合 $\{(x_i, y_i)\}_{i=1}^n$ が与えられたとき，各文 x の部分構造 x' をそれぞれ教師なしで抜き出し，全体として信頼度の高い知識 $\{(x'_i, y'_i)\}_{i=1}^n$ を出力する問題を考える．出力の信頼度は，PMI を一般化し，**確率的従属性**（依存性）によって判断する（2 節）．

スパースネスへの対応

さらに，スパースネスへの対応を考える．本研究では，従属性の尺度として Hilbert–Schmidt Independence Criterion (HSIC) [2] を用いる．HSIC はカーネル法ベースの独立性・従属性尺度であり，ほかの知識との類似性を考慮して知識の信頼度を計算することができる．たとえば〈have dinner, be full〉周辺文脈を考慮したスパースなイベント表現に対しても，言い換え表現や類義語を考慮して確率的従属性を計算することができる．また，組合せ最適化問題の解空間が広大であるため，MCMC で確率的に山登りをすることで探索コストを削減する（3 節）．また，実験により問題の定式化および提案手法が好ましい性質を持つことを示す．(4 節) 本研究の貢献は以下の通り：(1) 周辺文脈をケースバイケースで知識表現に組み込むため，粒度調整問題を定式化．(2) イベント間類似度を考慮した従属性尺度を用いることで，スパースな知識に対して信頼度を計算できる手法を提案．(3) 問題設定と提案手法の妥当性を確認．

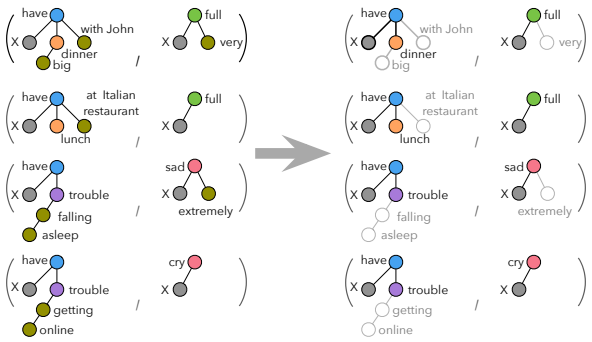

図 1: 粒度調整問題における望ましい入出力の例

2 粒度調整問題

前節で述べた問題の定式化をおこなう．入力として文のペアの集合 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ を受け取り，各 x_i, y_i から適当な部分構造（語の集合） x'_i, y'_i を抜き出し，信頼度の高い知識の集合 $\mathcal{D}' = \{(x'_i, y'_i)\}_{i=1}^n$ を教師なしで求めたい（図 1）．図 1 では，各文を依存構造木で表現した．

出力 $\mathcal{D}'$ の信頼度の高さはどのように測れば良いだろうか．典型的に，コーパスから獲得された関係知識 (x, y) の信頼度は PMI で計算され：

$$\mathrm{pmi}(x, y) = \log \frac{p(x, y)}{p(x)p(y)},$$

知識全体 $\{(x_i, y_i)\}_{i=1}^n$ の関連の良さは PMI の足し合わせである相互情報量で計算される：

$$\begin{aligned}\mathrm{MI}(X, Y) &= \sum_{(x,y)} p(x, y)\mathrm{pmi}(x, y) \\ &= \mathrm{KL}[P_{XY} \| P_X P_Y].\end{aligned}$$

言い換えれば，知識全体の信頼度を「$\mathcal{D}'$ を確率変数対 (X, Y) からの有限サンプルと見たときの X と Y の**従属性**の大きさ，P_{XY} と $P_X P_Y$ の離れ具合」をもって測っていることになる．本研究もこれに倣う．

以上より，解くべき最適化問題は以下の通り．

入力 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subseteq \mathcal{X} \times \mathcal{Y}$．ただし各 x_i, y_i は文であり，その表現（BoW，依存構造木，ベクトルなど）は任意．$\mathcal{X}$, $\mathcal{Y}$ は，それぞれ x_i, y_i およびその部分構造全体がなす集合．

出力 $\mathcal{D}' = \{(x'_i, y'_i)\}_{i=1}^n \subseteq \mathcal{X} \times \mathcal{Y}$．ただし各 i について $x'_i \subseteq x_i$, $y'_i \subseteq y_i$．‘$\subseteq$’ は適当な部分構造．

目的関数 出力 $\mathcal{D}'$ を確率変数対 (X, Y) からの有限サンプル：

$$\mathcal{D}' = \{(x'_i, y'_i)\}_{i=1}^n \sim_{\text{i.i.d.}} (X, Y)$$

と見たときの X と Y の**従属性**の大きさ $\to$ max. ただし従属性の尺度は任意．

3 提案手法：スパースネスへの対応

前節で挙げた問題を解くためには，(i) 文およびその部分構造の表現および (ii) 従属性の尺度をそれぞれ定める必要がある．本研究では，(i) 文の表現として依存構造木を，(ii) また従属性の尺度として HSIC [2] を採用した．HSIC は類似度をベースにした従属性尺度であり，「はじめに」で述べたスパースネスの問題によく適している．また，広大な解空間における探索については確率的な山登りを採用した．

3.1 文の表現：依存構造木

文や句 x_i, y_i は依存構造木で表現し，その部分構造 $x'_i \subseteq x_i$ は $y'_i \subseteq y_i$ は元の依存構造木の根付き部分木とした．依存構造木において，述語動詞や主語（のヘッド）や直接目的語（のヘッド）などのより強い情報が木のルートの近くにあり，修飾語句などのより仔細な情報が葉の側にあるため，より小さな根付き部分木がより抽象的な情報を表す．つまり，寝つき部分木のサイズを調整することで，知識の粒度の調整になるという直感に基づく．

3.2 従属性の尺度：HSIC

従属性の尺度としては，昨今多くのドメインで盛んに用いられているカーネル法ベースの独立性・従属性尺度である Hilbert–Schmidt Independence Criterion (HSIC) [2] を採用した．HSIC は非負値をとり，また確率変数対が独立であることとその値が 0 になることが必要十分である．独立性が下がるにしたがって，すなわち P_{XY} と $P_X P_Y$ が離れる[*1]にしたがって値が大きくなる．

HSIC の推定量は $\mathcal{D}' = \{(x'_i, y'_i)\}_{i=1}^n$ に対して以下の通り計算できる [2]：

$$\mathrm{HSIC}(\mathcal{D}', k, l) = \frac{1}{n^2}\mathrm{tr}(KHLH) = \frac{1}{n^2}\mathrm{tr}(\tilde{K}\tilde{L}). \tag{1}$$

ただし $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ および $l\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ は正定値カーネル[*2]，$K = (k(x'_i, x'_j)) \in \mathbb{R}^{n \times n}$ および $L = (l(y'_i, y'_j)) \in \mathbb{R}^{n \times n}$ はグラム行列，$\tilde{K} = HKH$ および $\tilde{L} = HLH$ は $H = ((\delta_{ij} - \frac{1}{n})) \in \mathbb{R}^{n \times n}$ を用いて計算できる中心化グラム行列[*3]である．

事態間関係知識獲得における HSIC のメリットは以下の 2 点が挙げられる．1 点目に，HSIC はノンパラメトリックな推論である．ニューラルネットワークをはじめとした各種手法のように，関数 $\mathcal{X} \to \mathcal{Y}$ をパラメトリックには考えていない．事態関係知識は個々のイベントに対してペアを組む相手が固定されるわけでは

[*1] $(\mathcal{X}, k)$, $(\mathcal{Y}, l)$ が定める再生核ヒルベルト空間をそれぞれ $\mathcal{H}_\mathcal{X}$, $\mathcal{H}_\mathcal{Y}$，$\mathcal{X}$ 上の確率変数 X のカーネル平均埋め込みを m_X としたときの，$\|m_{XY} - m_X m_Y\|^2_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{Y}}$ に対応する．

[*2] 直感的には文の部分構造間に定義される類似度．

[*3] 要素の値の平均を 0 にした（中心化した）グラム行列．

なく，関係知識 (x, y) を関数 $f : x \mapsto y$ で特徴付けるのは困難であると予想される．たとえば‘食事をする’というイベントに対して

- 〈食事をする，幸せになる〉
- 〈食事をする，会計をする〉
- 〈食事をする，太る〉

といった全く異なる関係があり得る．

2 点目に，HSIC は相互情報量と異なり，意味的な類似性を考慮してスムージングできる．たとえば $p_1 = \langle$have dinner, be full$\rangle$ と $p_2 = \langle$have lunch, be full$\rangle$ という二つの知識を考えたとき，PMI の観点では $\mathrm{pmi}(p_1, p_2) = 0$ となる，すなわち知識 p_2 によって知識 p_1 のスコアは強化されない．一方 HSIC を用い，‘have dinner’ と ‘have lunch’ の類似度を高く算出するような正定値カーネルを用いれば，「〈have lunch, be full〉という知識が周りにあったので〈have dinner, be full〉は良い知識と言えそうだ」という推論を暗黙的に入れることができる．数式上も，HSIC の推定量（式 1）は「$k(x'_i, x'_j)$ が大きければ $l(y'_i, y'_j)$ も大きく，$k(x'_i, x'_j)$ が小さければ $l(y'_i, y'_j)$ も小さい」という関係が各「ペアのペア」$((x'_i, y'_i), (x'_j, y'_j))$ に対してよく成り立つときに値が大きくなる．すなわち，各知識 (x'_i, y'_i) が，残りの知識 $\mathcal{D}' \setminus \{(x'_i, y'_i)\}$ と整合的であるとき HSIC は大きな値をとる．

3.3 カーネル

本研究では，$\mathrm{HSIC}(\mathcal{D}', k, l)$ の計算に用いる正定値カーネル k, l として，部分木 $\mathrm{Vec}(\cdot)$ を表すベクトル間のコサイン[*4]を用いた：

$$k(x'_i, x'_j) = \cos(\mathrm{Vec}(x'_i), \mathrm{Vec}(x'_j)),$$
$$l(y'_i, y'_j) = \cos(\mathrm{Vec}(y'_i), \mathrm{Vec}(y'_j)).$$

部分木 x を表すベクトル $\mathrm{Vec}(x)$ は，部分木を構成する単語のうち，学習済み単語ベクトルの語彙集合 Vocab に含まれており，かつストップワードの集合 SW に含まれていない単語について，これらを表す単語ベクトルたち $\mathrm{WordVecs}(x)$ の平均をとった：

$$\mathrm{WordVecs}(x) = \{\boldsymbol{v}(w) \mid w \in x, w \in \mathrm{Vocab}, w \notin \mathrm{SW}\},$$
$$\mathrm{Vec}(x) = \tfrac{1}{|\mathrm{WordVecs}(x)|} \textstyle\sum_{v_i \in \mathrm{WordVecs}(x)} v_i.$$

ストップワードの集合 SW は `http://www.ranks.nl/stopwords` より採取した．$|\mathrm{SW}| = 174$.

単語ベクトル $\boldsymbol{v}(\cdot)$ は，Google News より Skip-gram negative sampling [4] で作成した 300 次元の学習済みベクトルを用いた[*5]．単語ベクトルの語彙 Vocab の大きさは $|\mathrm{Vocab}| = 3 \times 10^6$.

[*4] $\cos\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ は正定値カーネルであり，HSIC の適用条件を満たす．

[*5] `https://code.google.com/archive/p/word2vec/`

3.4 探索アルゴリズム：MH

解くべき最適化問題（2 節）の解空間は広大である．すなわち各 x_i, y_i の部分構造の取り方のすべての組合せを探索するのは困難である．そこで本研究ではメトロポリス・ヘイティングス法 (MH) ベースの焼きなましにて最適化する．具体的には，「HSIC が大きい $\mathcal{D}'$ はより高い確率を持つ」ような $\mathcal{X} \times \mathcal{Y}$ 上の確率分布

$$p(\mathcal{D}'; \beta) \propto \exp(\beta \cdot \mathrm{HSIC}(\mathcal{D}', k, l)) \tag{2}$$

を考え，逆温度パラメータ β をあげながら分布 (2) 上で MCMC (MH) によるサンプリングを繰り返すことで，徐々に確率の大きな $\mathcal{D}'$，すなわち HSIC が大くなる $\mathcal{D}'$ を探索する．

探索アルゴリズムは以下の通り．

1. 現在の状態 $\mathcal{D}' = \{(x'_i, y'_i)\}_i$ から，構成する木をひとつ一様分布で選択：
$$\forall i,\ p(x'_i|\mathcal{D}') = p(y'_i|\mathcal{D}') = \frac{1}{2n}.$$
2. 選択された木 x' を構成するノード (単語) のうち，用いるか用いないかをスイッチしても「元の木の根付き部分木」という構造が維持されるノードの集合 $\mathrm{Switchable}(x')$ を考え，$\mathrm{Switchable}(x')$ の中から用いるか用いないかをスイッチする単語 x'' を一様分布で選択する：
$$Q(x''|x') = 1/|\mathrm{Switchable}(x')|.$$
$\mathcal{D}' = \{\ldots, (x', y'), \ldots\}$ から $x' \mapsto x''$ のみを変更し $\mathcal{D}'' = \{\ldots, (x'', y'), \ldots\}$ を選択する提案分布は次の通り：
$$Q(\mathcal{D}''|\mathcal{D}') = Q(x''|x')p(x'|\mathcal{D}') = \frac{1}{2n}Q(x''|x').$$
3. 以下の r を計算し，確率 $\min(1, r)$ で候補 $\mathcal{D}''$ を受理：
$$\begin{aligned} r &= \frac{p(\mathcal{D}''; \beta)}{p(\mathcal{D}'; \beta)} \cdot \frac{Q(\mathcal{D}'|\mathcal{D}'')}{Q(\mathcal{D}''|\mathcal{D}')} \\ &= \frac{\exp(\beta(\mathrm{HSIC}(\mathcal{D}'')))}{\exp(\beta(\mathrm{HSIC}(\mathcal{D}')))} \cdot \frac{Q(x'|x'')p(x''|\mathcal{D}'')}{Q(x''|x')p(x'|\mathcal{D}')} \\ &= \exp(\beta(\mathrm{HSIC}(\mathcal{D}'') - \mathrm{HSIC}(\mathcal{D}')))\frac{Q(x'|x'')}{Q(x''|x')}, \end{aligned}$$
ただし $\mathrm{HSIC}(\mathcal{D}') := \mathrm{HSIC}(\mathcal{D}', k, l)$.
4. 1〜3 を繰り返す．

4 実験

ここでは入力として文ペアを 12 組与え，出力を確認する．なお実際の実装は，データ数 10,000 程度の入力に対して学習が可能なことを確認済みである．ここでは提案手法の挙動が分かりやすいよう，少数の入力とその出力を確認する．

表 1 は実験の出力 $\mathcal{D}'$ を表したものであり，各行が各文ペアに対する出力 (x'_i, y'_i) を表す．たとえば 1 行目の ‘(They*) **had* breakfast** (at*) (the*) (eatery) (.*) — (They*) (are*) **full** (now) (.*)’ は，入力 (x_1, y_1) が (‘They had breakfast at the eatery.’, “They are full now.’) であり，出力 $(x'_1, y'_1) = \langle$**had breakfast**, **full**$\rangle$ が太字で表されている．丸括弧は出力に用いられなかった単語を示し，とくにアスタリスクが付いた単語は類似度計算から省略したストップワードである．

アルゴリズムが，類似性を考慮しながら共通部分を抜き出し，

- 上段の 4 ペアからは〈ご飯を食べる，腹がふくれる〉
- 中段の 4 ペアからは〈**友人と**ご飯を食べる，幸せな気持ちになる〉
- 下段の 4 ペアからは〈苦労がある，涙が出る〉

という知識が獲得されていることがわかる．

表 1: 人工データの実験の出力

x'_i — y'_i
(They*) **had* breakfast** (at*) (the*) (eatery) (.*) — (They*) (are*) **full** (now) (.*)
(I) (have*) **had* breakfast** (at*) (ten) (.*) — (I) (’m*) **full** (.*)
(We*) **had*** (special) **dinner** (.*) — (We*) (are*) **full** (.*)
(I) (have*) **had* breakfast** (at*) (my*) (house) (.*) — (I) (am*) **full** (.*)
(She*) **had* breakfast** (with*) (her*) **friends** (.*) — (She*) **felt 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* breakfast** (with*) (my*) **friends** (at*) (my*) **uncle** (’s) **house** (.*) — (I) **feel happy** (.*)
(He*) **had* trouble** (with*) (his*) (homework) (.*) — (He*) **cries** (.*)
(I) **had* trouble** (associating) (with*) (others*) (.*) — (I) **cry** (.*)
(She*) **had* trouble** (reading) (books) (.*) — (She*) **cries** (.*)
(I) **have* trouble** (concentrating) (.*) — (I) **cry** (.*)

実験結果から以下のことが読み取れる．

- 複数の入力に共通する部分構造，すなわち PMI を計算しても高くなるような部分構造は出力に含まれていることが見てとれる．たとえば第 1 ブロックの〈have breakfast, full〉は多くの入力に共通しており，これらは出力に含まれている．
- 限られた文にのみ登場する語は出力には含まれない．たとえば第 1 ブロックの ‘at my house’ は出力には含まれない．
- 類似性を考慮した従属性尺度の導入の効果が見てとれる．たとえば ‘dinner’ や ‘lunch’ はそれぞれ 1 度しか登場しない単語であるが，知識として残されている．これは ‘breakfast’ との類似性がカーネル関数により捉えられたものと考えられる．
- 第 2 ブロックの ‘(with) friends’ が知識に残されている．第 1 ブロックから得られる知識〈have breakfast, full〉と第 2 ブロックから得られる知識〈have breakfast (with) friends, feel happy〉が矛盾なく成立するためにアルゴリズムが ‘(with) friends’ を残したと考えられる．

5 おわりに

本稿ではテキスト集合からの事態間関係知識獲得において知識の粒度を文毎に調整する問題を定式化し，カーネル法ベースの従属性尺度 HSIC を利用した手法を提案した．HSIC の採用により，周辺文脈を考慮したスパースな知識に対して従属性を計算できるようになった．また，MH ベースの焼きなまし法により，スケーラビリティを確保した．さらに問題設定および提案手法の妥当性を人工的な小規模データで確認した．

本稿で取り上げた挙げた粒度調整の問題は，事態間関係知識の獲得のみならず自然言語処理の内外に広くあらわれる．たとえば，動詞と項の関連の良さを考える選択選好の問題，OpenIE の問題などは，ペアを構成する要素のスパンやセグメントをインスタンス毎に決定するのが困難であり，今回のアプローチが有効であると考えられる．

また，カーネル法であるため計算量の問題が懸念されるが，データ数 n に対して素朴には $O(n^2)$ のコストがかかるグラム行列の構成は最初の 1 回だけであり，MH の 1 iteration 毎にはグラム行列の 1 行分の更新だけで良く $O(n)$ の再計算で十分である．実際，Python による実装でもデータ数 10,000 程度で学習できることを確認できている．さらなる高速化手法として，不完全コレスキー分解を用いたグラム行列の近似なども考えられる．

謝辞 本研究は JST CREST の支援を受けた．

参考文献

[1] Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Event Chains. In *ACL*, pp. 789–797, 2008.
[2] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring Statistical Dependence with Hilbert-Schmidt Norms. In *ALT*, pp. 63–77, 2005.
[3] Mehdi Manshadi, Reid Swanson, and Andrew S Gordon. Learning a Probabilistic Model of Event Sequences From Internet Weblog Stories. In *FLAIRS*, pp. 159–164, 2008.
[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, pp. 1–9, 2013.
[5] Karl Pichotta and Raymond Mooney. Statistical Script Learning with Multi-Argument Events. *EACL*, pp. 220–229, 2014.