Learning Co-Substructures by Kernel Dependence Maximization

Sho Yokoi [1], **Daichi Mochihashi** [2], **Ryo Takahashi** [1], **Naoaki Okazaki** [1], **Kentaro Inui** [1]
[1] Tohoku University, Sendai, Japan
[2] The Institute of Statistical Mathematics, Tokyo, Japan
{yokoi, ryo.t, okazaki, inui}@ecei.tohoku.ac.jp, daichi@ism.ac.jp

Abstract

Modeling associations between items in a dataset is a problem that is frequently encountered in data and knowledge mining research. Most previous studies have simply applied a predefined fixed pattern for extracting the substructure of each item pair and then analyzed the associations between these substructures. Using such fixed patterns may not, however, capture the significant association. We, therefore, propose the novel machine learning task of extracting a strongly associated substructure pair (co-substructure) from each input item pair. We call this task *dependent co-substructure extraction (DCSE)*, and formalize it as a dependence maximization problem. Then, we discuss critical issues with this task: the data sparsity problem and a huge search space. To address the data sparsity problem, we adopt the Hilbert–Schmidt independence criterion as an objective function. To improve search efficiency, we adopt the Metropolis–Hastings algorithm. We report the results of empirical evaluations, in which the proposed method is applied for acquiring and predicting narrative event pairs, an active task in the field of natural language processing.

1 Introduction

Modeling associations between items in a dataset is a general problem commonly addressed in a broad range of data or knowledge mining research. For example, the valuable natural language processing tasks of extracting narrative event pairs (e.g., ⟨*X commit a crime*, *X be arrested*⟩) as components of script knowledge [Chambers and Jurafsky, 2008] and learning selectional preference of predicates (e.g., *food* is preferred as an object of *eat*) [Resnik, 1997] model associations between event pairs and predicate-argument pairs, respectively.

The common approach for modeling associations usually involves three steps; we use the approach proposed by Chambers and Jurafsky (C&J) as an example. In the first step, associated pairs of items are collected from a dataset as positive samples; in the C&J method, these are sentence pairs that include co-referent people or objects from a text corpus (e.g., the sentence pair "*Tom$_i$ killed Nancy.*" and "*The police arrested Tom$_i$ immediately.*" is collected because of the co-referent "*Tom$_i$*"

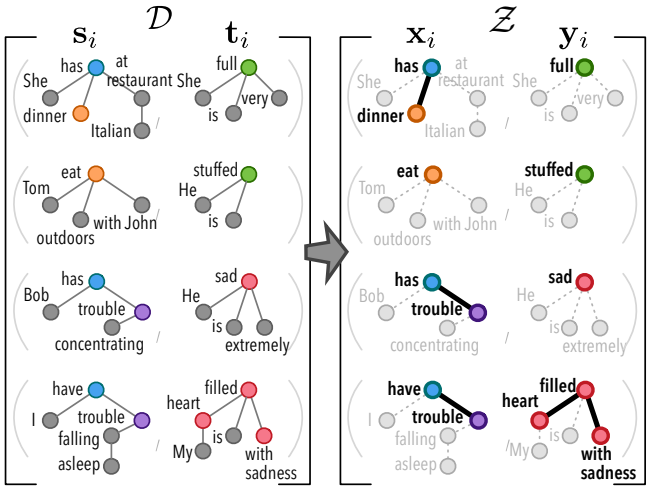


Figure 1: Example of the input and output of Dependent Co-Substructure Extraction (DCSE) for acquiring narrative event pairs.

in both the sentences). In the second step, the abstract representation (substructure) is extracted from each item pair; the C&J method utilizes head predicates coupled with argument slots (e.g., ⟨*X kill*, *arrest X*⟩). In the third step, the association between the extracted substructure pairs is modeled; the C&J method utilizes pointwise mutual information (PMI) to measure the association. In this way, ⟨*X kill*, *arrest X*⟩ (for example) might be chosen as a narrative event pair because the calculated PMI value is large; in contrast, event pairs with low PMI values such as ⟨*X kill*, *X graduate*⟩ would be discarded.

We believe the second step mentioned above can be improved because, in most previous studies, people simply applied intuitively defined and fixed patterns for extracting substructures, without much optimization for the third step. For example, the C&J method uses a simple syntactic pattern, namely one predicate with a co-referent argument slot, such as *X kill*. With such a fixed pattern, however, the intended associations between item pairs may not be best captured. Fig. 1 shows an example of this, where we assume that we are mining narrative event pairs from the set $\mathcal{D}$ of sentence pairs. If we were to use the syntactic pattern of the C&J's method, we would obtain event pairs such as ⟨*X have*, *X full*⟩ and ⟨*X have*, *X sad*⟩. However, the event representation *X have* is clearly too abstract to capture the full associations with the distinct

events *X full* and *X sad*. Ideally, we want to acquire event pairs such as ⟨*X have dinner*, *X full*⟩ and ⟨*X have trouble*, *X sad*⟩ as illustrated by the set $\mathcal{Z}$ in Fig. 1. Therefore, ideally, we should be able to flexibly choose substructures of arbitrary size that are the most appropriate to capture the associations.

This issue motivates us to consider a new machine learning task, which we call *dependent co-substructure extraction (DCSE)*. In this task, a pair of strongly associated substructures $\langle \mathbf{x}_i, \mathbf{y}_i \rangle$ are extracted for each input item pair $\langle \mathbf{s}_i, \mathbf{t}_i \rangle$, as illustrated in Fig. 1. We consider selecting an appropriate level of knowledge abstraction to mine based on the association strength. In Fig. 1, for example, we want to include *dinner* in the substructure of the first pair but not *at restaurant*, using no predefined pattern.

In this paper, we first formalize the task of DCSE as a dependence maximization problem; then, we discuss two critical issues with the task: the data sparsity problem and the huge search space (Sec. 2). We propose adopting the Hilbert–Schmidt independence criterion (HSIC) as an objective function to cope with the data sparsity problem (Sec. 3) and the Metropolis-Hastings (MH) algorithm to boost search efficiency (Sec. 4). Finally, we demonstrate the superiority of the proposed method via experiments in two scenarios, namely knowledge acquisition and predicting narrative event pairs (Sec. 5).

2 DCSE as Dependence Maximization

First, we formalize our dependence maximization problem. For each given pair of items, find a strongly associated pair of substructures based on the dependence maximization principle as follows.

Dependent co-substructure extraction (DCSE).
Given: A set $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{t}_i)\}_{i=1}^N$ of item pairs, where $\mathbf{s}_i \in \mathcal{S}$ and $\mathbf{t}_i \in \mathcal{T}$ are raw items and each item pair $(\mathbf{s}_i, \mathbf{t}_i)$ represents a specific relation of interest (e.g., co-reference and co-occurrence). $\mathcal{S}$ and $\mathcal{T}$ are sets containing all the raw data $\{\mathbf{s}\}$ and $\{\mathbf{t}\}$, respectively.
Find: A set $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ of substructure pairs (co-substructures) that maximizes the dependence (given below), where $\mathbf{x}_i \preccurlyeq \mathbf{s}_i$, $\mathbf{y}_i \preccurlyeq \mathbf{t}_i$ for each i and '$\mathbf{x} \preccurlyeq \mathbf{s}$' denotes that $\mathbf{x}$ is a substructure of $\mathbf{s}$. To estimate the dependence, we assume that each solution $\mathcal{Z}$ as N independent samples drawn from some joint distribution:

$$\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \sim P_{XY}. \quad (1)$$

Then, we estimate the dependence between X and Y by the distance between the joint density P_{XY} and the product of marginals $P_X P_Y$.

There are several possible ways of measuring the distance between P_{XY} and $P_X P_Y$; one straightforward way is to employ mutual information (MI) as a dependence measure. MI has been utilized various knowledge mining and machine learning studies to measure association strength and dependence [Church and Hanks, 1990; Maes *et al.*, 1997; Turney, 2002; Torkkola, 2003; Peng *et al.*, 2005]. Using MI, the dependence of $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ is computed by

$$\mathrm{MI}(\mathcal{Z}) = \sum_{\mathbf{x}} \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}) p(\mathbf{y})} \quad (2)$$

$$= \mathrm{KL}[P_{XY} \| P_X P_Y], \quad (3)$$

where $\mathrm{KL}[\cdot\|\cdot]$ denotes the Kullback–Liebler divergence between two distributions.

Adopting MI in DCSE dose, however, poses two critical problems:

Data sparsity Our search space includes substructure pairs of arbitrary size. In the case of Fig. 1, for example, we may consider a specific substructure such as "*She has big dinner*" as a candidate substructure. Thus, we encounter a data sparsity problem if the probability distribution of substructures is naïvely estimated by counting occurrences in the data.

Huge search space The search space of DCSE can be prohibitively large. The optimal co-substructure for a given input item pair depends on the co-substructure choices for other item pairs, making searching difficult. In other words, one cannot reach the global optimal simply by locally choosing seemingly good co-substructures. There is, therefore, a need to devise an approximation method to improve search efficiency.

We propose a solution to these problems in the subsequent two sections.

3 Objective Function: HSIC

To cope with the data sparsity problem, we propose adopting the Hilbert–Schmidt independence criterion (HSIC) [Gretton *et al.*, 2005], instead of MI, as the objective function. HSIC is a kernel-based dependence measure involving low computational cost that has been used in a range of machine learning tasks such as feature selection [Song *et al.*, 2012], dimensionality reduction [Fukumizu *et al.*, 2009], and unsupervised object matching [Quadrianto *et al.*, 2009].

Intuitively, HSIC can be seen as a smoothed version of MI, where the candidate substructure counts are somehow smoothed using similarities between substructures. Consider the example in Fig. 1 again. Using HSIC, the (semantic) similarity between the substructures "*have dinner*" and "*eat*" can be considered when estimating the dependence of $\mathcal{Z}$, which smoothens the counts of those substructures.

Let X and Y be random variables with ranges $\mathcal{X}$ and $\mathcal{Y}$ respectively (i.e. $\mathcal{X}$ constitutes all candidate substructures $\{\mathbf{x} \colon \mathbf{x} \preccurlyeq \mathbf{s}, \mathbf{s} \in \mathcal{S}\}$, and similarly for $\mathcal{Y}$) and $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \subseteq \mathcal{X} \times \mathcal{Y}$ be a series of N independent observations drawn from the joint distribution P_{XY}. The HSIC value[1] of $\mathcal{Z}$, which estimates the degree of dependence between X and Y, is

$$\mathrm{HSIC}(\mathcal{Z}; k, \ell) = \frac{1}{N^2} \mathrm{tr}(\mathbf{KHLH}) = \frac{1}{N^2} \mathrm{tr}(\tilde{\mathbf{K}}\tilde{\mathbf{L}}). \quad (4)$$

In this equation,

- $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\ell \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ are positive definite kernels that serve as similarity functions over substructures,

[1] Precisely, HSIC measures the dependence between two random variables, X and Y. Eq. 4 gives an empirical estimator of HSIC.

- $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j)) \in \mathbb{R}^{N \times N}$ and $\mathbf{L} = (\ell(\mathbf{y}_i, \mathbf{y}_j)) \in \mathbb{R}^{N \times N}$ are Gram matrices, which serve as the similarity matrices given by kernel functions k and l, and
- $\tilde{\mathbf{K}} = \mathbf{HKH} \in \mathbb{R}^{N \times N}$ and $\tilde{\mathbf{L}} = \mathbf{HLH} \in \mathbb{R}^{N \times N}$ are centered Gram matrices, where $\mathbf{H} = ((\delta_{ij} - \frac{1}{N})) \in \mathbb{R}^{N \times N}$.

To elaborate on the intuition that the HSIC is a smoothed version of MI, let us first consider a smoothed version of PMI. We define *pointwise HSIC (PHSIC)*: $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ with given $\mathcal{Z}$, as follows:

$$\mathrm{PHSIC}(\mathbf{x}, \mathbf{y}; \mathcal{Z}) := \sum_{i=1}^{N} \tilde{k}(\mathbf{x}, \mathbf{x}_i; \{\mathbf{x}_n\}) \tilde{\ell}(\mathbf{y}, \mathbf{y}_i; \{\mathbf{y}_n\}), \quad (5)$$

where $\{\mathbf{x}_n\}$ denotes $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. The function $\tilde{k}(\cdot, \cdot; \{\mathbf{x}_n\}) \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined by

$$\begin{aligned} \tilde{k}(\mathbf{x}, \mathbf{x}'; \{\mathbf{x}_n\}) := k(\mathbf{x}, \mathbf{x}') - \frac{1}{N} \sum_{j=1}^{N} k(\mathbf{x}, \mathbf{x}_j) \\ - \frac{1}{N} \sum_{i=1}^{N} k(\mathbf{x}_i, \mathbf{x}') + \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} k(\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \quad (6)$$

which gives the similarity between $\mathbf{x}$ and $\mathbf{x}'$ centered in future space; namely, if addition is defined on $\mathcal{X}$, $\tilde{k}(\mathbf{x}, \mathbf{x}'; \{\mathbf{x}_n\})$ equals to $\tilde{k}(\mathbf{x} - \bar{\mathbf{x}}_n, \mathbf{x}' - \bar{\mathbf{x}}_n)$. The PHSIC value of $\mathbf{x}$ and $\mathbf{y}$ is essentially increased by the presence of other samples $(\mathbf{x}_i, \mathbf{y}_i)$ that are similar to $(\mathbf{x}, \mathbf{y})$, i.e., $\mathbf{x}_i \approx \mathbf{x}$ and $\mathbf{y}_i \approx \mathbf{y}$; this enables smoothing across similar items. Moreover, HSIC corresponds to the summation of PHSIC values, paralleling the relationship between MI and PMI (i.e., MI is the summation of PMI values.) This is the sense in which HSIC can be seen as a smoothed version of MI. The relationship between MI and HSIC is summarized in Table 1.

This smoothing is a strong advantage of the HSIC. In knowledge mining and natural language processing, a wide range of methods can be used for estimating similarities between words, phrases, and their arbitrary substructures, from classical thesaurus-based methods to modern embedding-based ones. Using such a similarity function, one can consider, for example, the co-occurrence $\langle \underline{eat}, \textit{full} \rangle$ when estimating the association strength of, for example, $\langle \underline{\textit{have dinner}}, \textit{full} \rangle$. By solving the HSIC maximization problem, strongly associated co-substructures of arbitrary size can be extracted while coping with the data sparsity problem.

4 Search: Based on Metropolis–Hastings

In order to find a near-optimal $\mathcal{Z}$ in a huge search space, we adopt an approach based on Metropolis–Hastings (MH) sampling [Chib and Greenberg, 1995]. We consider the probability distribution

$$p(\mathcal{Z}; k, \ell, \beta) \propto \exp(\beta \cdot \mathrm{HSIC}(\mathcal{Z}; k, \ell)), \quad (7)$$

where β is the inverse of a temperature parameter. The larger an HSIC value is, the higher $\mathcal{Z}$'s probability. By sampling $\mathcal{Z}$ on the distribution given by Eq. 7 while changing $\mathcal{Z}$ step by step, $\mathcal{Z}$ is expected to converge to its optimal value with a substantially lower computational cost than that with a full search (Fig. 2).

The details of the sampling procedure are as follows.

1. Let $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ be the current sample.

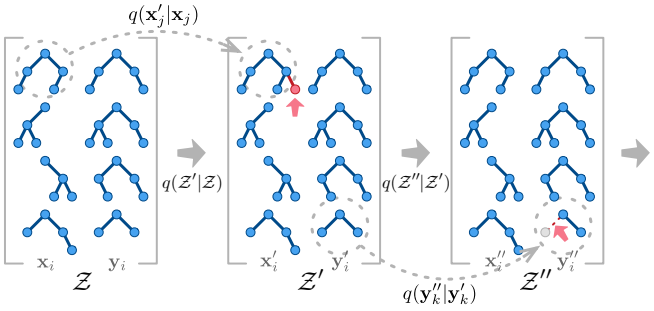


Figure 2: Overview of Metropolis–Hastings sampling.

2. Draw a new candidate $\mathcal{Z}'$ by changing only one substructure. Specifically, first, draw $\mathbf{x}_i$ or $\mathbf{y}_i$ from $\mathcal{Z} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ from a uniform distribution: $\forall i,\ p(\mathbf{x}_i|\mathcal{Z}) = p(\mathbf{y}_i|\mathcal{Z}) = \frac{1}{2N}$. Then, propose an $\mathbf{x}'_i$ for a given $\mathbf{x}_i$ using a proposal distribution $q(\mathbf{x}'_i|\mathbf{x}_i)$ (a specific example of $q(\mathbf{x}'_i|\mathbf{x}_i)$ is given in Sec. 5.1). Thus, the proposed distribution for drawing a new candidate $\mathcal{Z}' = \{\ldots, (\mathbf{x}'_i, \mathbf{y}_i), \ldots\}$ for a given $\mathcal{Z} = \{\ldots, (\mathbf{x}_i, \mathbf{y}_i), \ldots\}$ by changing only $\mathbf{x}_i$ is
$$q(\mathcal{Z}'|\mathcal{Z}) = q(\mathbf{x}'_i|\mathbf{x}_i) p(\mathbf{x}_i|\mathcal{Z}) = \frac{1}{2N} q(\mathbf{x}'_i|\mathbf{x}_i). \quad (8)$$
3. Accept $\mathcal{Z}'$ with the probability $\min(1, r)$, where,
$$r = \frac{p(\mathcal{Z}'; k, \ell, \beta)}{p(\mathcal{Z}; k, \ell, \beta)} \cdot \frac{q(\mathcal{Z}|\mathcal{Z}')}{q(\mathcal{Z}'|\mathcal{Z})} \quad (9)$$
$$= \exp(\beta(\mathrm{HSIC}(\mathcal{Z}'; k, \ell) - \mathrm{HSIC}(\mathcal{Z}; k, \ell))) \frac{q(\mathbf{x}|\mathbf{x}')}{q(\mathbf{x}'|\mathbf{x})}. \quad (10)$$
4. Repeat Steps 2–3.

Because the HSIC is a kernel-based measure, its high computational cost may be a concern. In reality, we only need to compute Gram matrices with $O(N^2)$ computational cost only for the first iteration. For each iteration of MH sampling, it is sufficient to update only one row of the Gram matrix with $O(N)$ computational cost. In addition, computing the HSIC takes only $O(N\kappa^2)$ time via rank κ incomplete Cholesky decomposition [Gretton *et al.*, 2005] (Lemma 2).

5 Experiments

We evaluated the performance of the proposed method in two scenarios.

- **Knowledge acquisition** extracts a set of abstract representation pairs, $\mathcal{Z} = \{(\mathbf{x}, \mathbf{y})\}$, from a corpus. We feed an input $\mathcal{D} = \{(\mathbf{s}, \mathbf{t})\}$ to a DCSE method, and then verify if the output $\mathcal{Z}$ is reasonable and interpretable. In order to examine the behavior of the proposed method, we perform knowledge acquisition on a synthetic dataset in Sec. 5.2.
- **Prediction** constructs a model from $\mathcal{Z}$, and computes the relevance of a new pair $(\mathbf{s}, \mathbf{t})$ based on the relevance of the substructure pairs $(\mathbf{x}, \mathbf{y})$. In other words, the determination of whether a new pair $(\mathbf{s}, \mathbf{t})$ has a relationship is based on the score of its abstract representation $(\mathbf{x}, \mathbf{y})$. We report the results of experiments on real corpora in Sec. 5.3.

5.1 Experimental Settings

Data Representation

We adopted dependency trees for the input representations of input $\mathbf{s}_i$ and $\mathbf{t}_i$ and their rooted subtrees as those for output

Table 1: Relationship between MI and HSIC. "$\veebar$" denotes exclusive or. When the equation indicated by the "+" sign is satisfied, the PMI/PHSIC value increases. When the equation indicated by the "−" sign is satisfied, the PMI/PHSIC value decreases. $\mathbb{I}[\text{cond}] = 1$ if the condition is true and 0 otherwise. Note that the elements of the centered Gram matrix $\tilde{\mathbf{K}}$ can be expressed as $\tilde{\mathbf{K}}_{ij} = \tilde{k}(\mathbf{x}_i, \mathbf{x}_j; \{\mathbf{x}_n\})$ and $\mathrm{HSIC}(\mathcal{Z}) = \frac{1}{N^2}\sum_{ij}\tilde{\mathbf{K}}_{ij}\tilde{\mathbf{L}}_{ij}$.

		consistency of $(\mathbf{x}, \mathbf{y})$ with $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{Z}$	consistency of $(\mathbf{x}, \mathbf{y})$ with $\mathcal{Z}$	estimate of dependency
MI	+	$\mathbf{x}=\mathbf{x}_i \wedge \mathbf{y}=\mathbf{y}_i$	$\mathrm{PMI}(\mathbf{x}, \mathbf{y}; \mathcal{Z}) = \log \frac{N \cdot \sum_i \mathbb{I}[\mathbf{x}=\mathbf{x}_i \wedge \mathbf{y}=\mathbf{y}_i]}{\sum_i \mathbb{I}[\mathbf{x}=\mathbf{x}_i]\sum_i \mathbb{I}[\mathbf{y}=\mathbf{y}_i]}$	$\mathrm{MI}(\mathcal{Z}) = \frac{1}{N}\sum_i \mathrm{PMI}(\mathbf{x}_i, \mathbf{y}_i)$
	−	$\mathbf{x}=\mathbf{x}_i \veebar \mathbf{y}=\mathbf{y}_i$		
HSIC	+	$\tilde{k}(\mathbf{x}, \mathbf{x}_i; \{\mathbf{x}_n\})\tilde{\ell}(\mathbf{y}, \mathbf{y}_i; \{\mathbf{y}_n\}) > 0$	$\mathrm{PHSIC}(\mathbf{x}, \mathbf{y}; \mathcal{Z}) = \sum_i \tilde{k}(\mathbf{x}, \mathbf{x}_i; \{\mathbf{x}_n\})\tilde{\ell}(\mathbf{y}, \mathbf{y}_i; \{\mathbf{y}_n\})$	$\mathrm{HSIC}(\mathcal{Z}) = \frac{1}{N^2}\sum_i \mathrm{PHSIC}(\mathbf{x}_i, \mathbf{y}_i)$
	−	$\tilde{k}(\mathbf{x}, \mathbf{x}_i; \{\mathbf{x}_n\})\tilde{\ell}(\mathbf{y}, \mathbf{y}_i; \{\mathbf{y}_n\}) < 0$		

substructures $\mathbf{x}_i$ and $\mathbf{y}_i$ (Fig. 1).

Kernel Function

HSIC requires two positive definite kernels, k and ℓ (Eq. 4), that compute the similarity between two subtrees of dependency trees. We employed the cosine similarity[2] between vector representations, which has several desirable properties: (i) it is the most standard function used to measure similarity between phrases using word vectors; (ii) it has no hyperparameter; and (iii) its computation cost is low.

Let $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ be rooted subtrees. We then defined a kernel function k,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \cos(\mathbf{v}_{\mathrm{tree}}(\mathbf{x}_i), \mathbf{v}_{\mathrm{tree}}(\mathbf{x}_j)), \quad (11)$$

and another kernel function ℓ analogously. The vector $\mathbf{v}_{\mathrm{tree}}(\mathbf{x})$ is the average of the word vectors for the word set $\{w\}$ constituting the subtree $\mathbf{x}$:

$$\mathbf{v}_{\mathrm{tree}}(\mathbf{x}) = \mathrm{average}(\{\mathbf{v}(w) \colon w \in \mathbf{x},\ w \in V\}). \quad (12)$$

Here, $\mathbf{v}(\cdot)$ denotes 300-dimensional pre-trained word vectors[3], and V represents a vocabulary set[4].

If the product kernel $k(\cdot,\cdot) \times \ell(\cdot,\cdot)$ is characteristic on $\mathcal{X} \times \mathcal{Y}$, $\mathrm{HSIC}(X, Y, k, \ell) = 0$ if and only if "X and Y are independenct" [Muandet *et al.*, 2016]. Therefore, when we test independence using HSIC, the kernels should be characteristic (e.g., Gaussian kernel and Laplacian kernel). However, in this study, we are more interested in the case of dependence (the HSIC value is large), rather than independence (the HSIC value is small). Therefore, the HSIC can be sufficiently estimated by only considering low-order moments of the probability distribution; this little negative effect even when using characteristic kernels. In fact, replacing the cosine kernel with the Gaussian kernel had almost no impact in our experiments (with the same setup otherwise).

Proposal Distribution

MH sampling uses a proposal distribution $q(\mathbf{x}'|\mathbf{x})$ that suggests a next candidate $\mathbf{x}' \preccurlyeq \mathbf{s}$ given the current candidate $\mathbf{x} \preccurlyeq \mathbf{s}$ (Eq. 8) and similarly uses $q(\mathbf{y}'|\mathbf{y})$ for $\mathbf{y}'$. For a given $\mathbf{x} \preccurlyeq \mathbf{s}$, let $M(\mathbf{x})$ be the set of subtrees of $\mathbf{s}$ that are obtained by stretching or shrinking only one edge of $\mathbf{x}$. The experiments used $q(\mathbf{x}'|\mathbf{x})$ that yielded $1/|M(\mathbf{x})|$ if $\mathbf{x}' \in M(\mathbf{x})$ and 0 otherwise (Fig. 3).

[2] $\cos(\cdot,\cdot) \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a positive definite kernel which satisfies the application condition of HSIC.

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

[4] Stop words in `http://www.ranks.nl/stopwords/` are excluded.

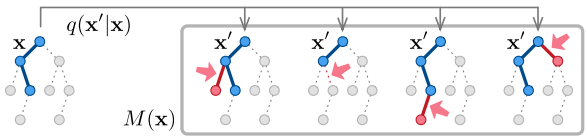


Figure 3: Proposal distribution $q(\mathbf{x}'|\mathbf{x})$ used in experiments.

5.2 Knowledge Acquisition from Synthetic Data

In order to verify that the proposed method yields reasonable and interpretable paired abstract representations, we prepared a small synthetic dataset constituting 12 pairs of sentences.

Results

Fig. 4 shows the experimental results. The upper half shows the input $\mathcal{D}$ and the lower half shows the output $\mathcal{Z}$ obtained via the proposed method. For example, the method abstracted the first input $(\mathbf{s}_1, \mathbf{t}_1) = \langle$*They had breakfast at the eatery.*, *They are full now.*$\rangle$ to $(\mathbf{x}_1, \mathbf{y}_1) = \langle$*had breakfast*, *full*$\rangle$ (number 1 in the figure). The heat maps in Fig. 4 represent centered Gram matrices. For example, the bottom-left heat map shows the similarity matrix $\tilde{\mathbf{K}}$ for $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$; the element at $(1, 9)$ represents the value of $\tilde{k}(\mathbf{x}_1, \mathbf{x}_9) = \tilde{k}($*had breakfast*, *had trouble*$)$.

Discussion

The proposed method successfully found the common substructures in the inputs. For example, the method recognized the co-substructure $\langle$*have breakfast*, *full*$\rangle$ in the first block because it was common to many inputs. By contrast, the method pruned rare words that were unimportant for modeling association. For example, the method removed "*at my house*" in $\mathbf{s}_4$ from the first block.

In addition, the proposed method works flexibly on surface variations by considering word similarity. Although the words "*dinner*" (in $\mathbf{s}_4$) and "*lunch*" (in $\mathbf{s}_7$) appeared only once in the corpus, the method found an appropriate common co-substructure by recognizing that they are similar to "*breakfast*". The method ultimately recognized three clusters in the input data—$\langle$*eat meals*, *be full*$\rangle$, $\langle$*eat meals with friends*, *feel happy*$\rangle$, and $\langle$*have trouble*, *cry*$\rangle$— from the upper, middle, and lower four pairs, respectively. We can also confirm this behavior by observing the $\tilde{\mathbf{K}}$ value (the bottom-left heat map in Fig. 4): the events in the first block and those in the second block in $\tilde{\mathbf{K}}$ are strongly similar, respectively (the squares surrounded by red lines in Fig. 4).

Furthermore, we can observe that "*(with) friends*" remained in the abstract representation in the second block. We infer that

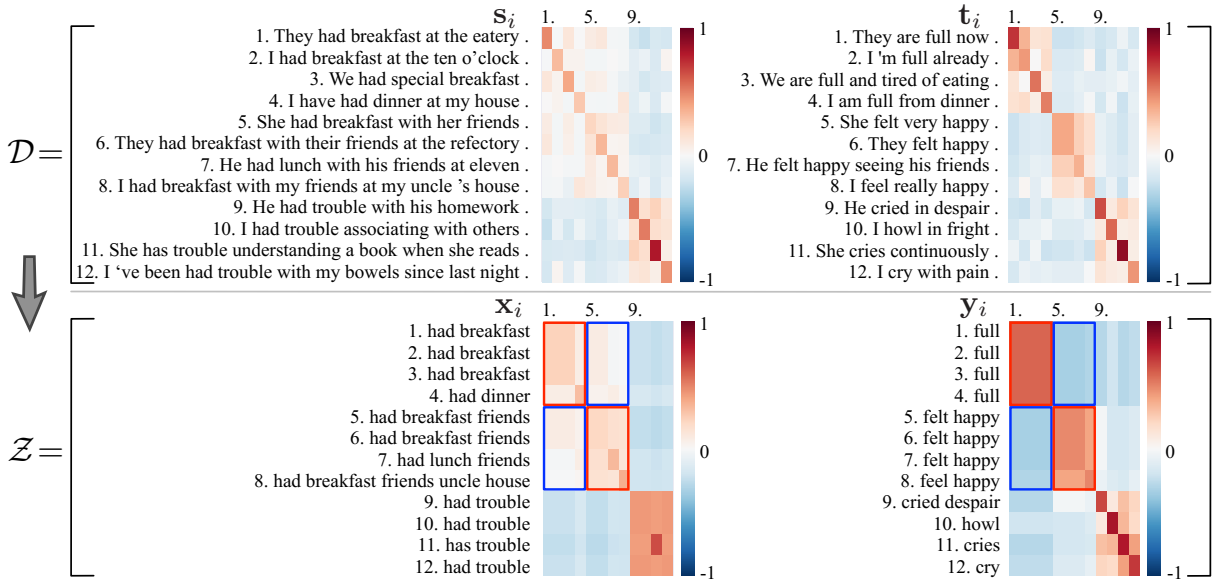


Figure 4: Results obtained via the knowledge acquisition task. The upper and lower halves respectively show the input $\mathcal{D}$ and the output $\mathcal{Z}$. The heat maps represent centered Gram matrices.

this is because removing the expression from the second block would result in merging of the first and second blocks on the $\mathcal{X}$ side, whereas the information in the first (*full*) and second (*felt happy*) blocks on the $\mathcal{Y}$ side was totally different. This merger would have been an undesirable behavior, decreasing the ability to predict $\mathbf{t}$ from $\mathbf{s}$. The proposed method prevented this behavior by observing the Gram matrices of $\mathcal{X}$ and $\mathcal{Y}$; the red and blue frames in Fig. 4 suggest that the first and second blocks are not merged.

5.3 Prediction on Real Corpora

In order to demonstrate the effectiveness of the proposed method on a real dataset and task, we conducted an experiment on pairwise classification of narrative event pairs. We first learned an association model with positive paired data gathered from corpora and then measured the model's prediction performance on a test dataset.

Dataset

Table 2 provides the data statistics for the performed prediction task. We used the following two corpora:

- **The Gigaword Corpus**[5] [Graff and Cieri, 2003]: a large collection of English newswire text data that has been used in several previous studies [Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009; Granroth-Wilding and Clark, 2016]. We used 17,781 documents published in the year 2000 from the New York Times (NYT) portion.
- **Andrew Lang Fairy Tale Corpus**[6]: a small collection of children's stories that has been used in a previous study [Jans *et al.*, 2012]. We used all 437 stories in this experiment.

Applying Stanford CoreNLP Version 3.7.0 [Manning *et al.*, 2014] to raw text from the corpora, we extracted sentence pairs sharing co-referring arguments. When handling the Gigaword and the Fairy Tale corpora, used *regular bigrams* and *2-skip bigrams*, respectively [Jans *et al.*, 2012].

Table 2: Data statistics for the prediction task.

Corpus	Collection	#All	#Training	#Test(pos)	#Test(neg)
Gigaword	regular	16,748	10,000	500	500
Fairy Tale	2-skip	1,673	1,000	100	100

Next, we filtered the sentence pairs using the following conditions:

- $4 \leq$ the number of tokens in a sentence ≤ 30;
- the POS tag of the root node of the dependency tree is in the set {VB, VBD, VBG, VBN, VBP, VBZ};
- the word at the root node of the dependency tree is not in the set {be, am, are, is, was, were, 'm, 're, 's}; and
- the position of all protagonists seen from the predicate verb are in the set {nsubj, dobj}.

We collected all sentences satisfying the above conditions into a set of positive sentence pairs $\mathcal{D}^{(\mathrm{all})}$.

Finally, we randomly chose positive sentence pairs from this set to construct the training set and the test set $\mathcal{D}_{\mathrm{P}}^{(\mathrm{test})}$ (without overlap). We obtained pseudo-negative sentence pairs $\mathcal{D}_{\mathrm{N}}^{(\mathrm{test})} = \{(\mathbf{s}', \mathbf{t}')\}$ for the test set by randomly extracting $\mathbf{s}'$ and $\mathbf{t}'$ from positive sentence pairs $\mathcal{D}^{(\mathrm{all})} = \{(\mathbf{s}, \mathbf{t})\}$.

Performance Measure: AUC

We used the area under the receiver operating characteristic curve (AUC-ROC or AUC) to evaluate the performance of the different scoring functions. This task is a pairwise binary classification problem and is essentially a version of the conventional "cloze test" for narrative event chains. In binary classification/ranking problems, AUC-ROC is generally used as an evaluation metric, and it is a stable and robust measure even when the ratio of positive and negative examples in the

[5] https://catalog.ldc.upenn.edu/ldc2003t05/
[6] http://www.mythfolklore.net/andrewlang/

test set is skewed, unlike the area under the precision-recall curve (AUC-PR) [Fawcett, 2006].

Given a set of positive examples $\mathcal{D}_{\mathrm{P}} = \{(\mathbf{s}, \mathbf{t})\}$ and a set of negative examples $\mathcal{D}_{\mathrm{N}} = \{(\mathbf{s}', \mathbf{t}')\}$, the AUC can be computed using any scoring function $f : \mathcal{S} \times \mathcal{T} \to \mathbb{R}$ as,

$$\frac{1}{|\mathcal{D}_{\mathrm{P}}||\mathcal{D}_{\mathrm{N}}|} \sum_{(\mathbf{s},\mathbf{t})\in\mathcal{D}_{\mathrm{P}}} \sum_{(\mathbf{s}',\mathbf{t}')\in\mathcal{D}_{\mathrm{N}}} \mathbb{I}[f(\mathbf{s},\mathbf{t}) > f(\mathbf{s}',\mathbf{t}')], \quad (13)$$

where $\mathbb{I}[\mathrm{cond}] = 1$ if the condition is true and 0 otherwise.

Experimental Procedure

Here, we explain the generic procedure for computing AUCs for both the proposed and baseline methods.

Training Train a model $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.

1. Abstraction: generate abstract event pairs $\mathcal{Z}$ for a given training set $\mathcal{D}^{(\mathrm{train})}$.
2. Training: construct an association model g between abstract representations from $\mathcal{Z}$.

Test Compute the score $f(\mathbf{s}, \mathbf{t})$ for each $(\mathbf{s}, \mathbf{t})$ in the test set.

1. Abstraction: convert the given pair $(\mathbf{s}, \mathbf{t})$ to an abstract representation $(\mathbf{x}, \mathbf{y})$ using the method in question.
2. Scoring: compute the score $g(\mathbf{x}, \mathbf{y})$ and regard it as the score for the original representation $f(\mathbf{s}, \mathbf{t})$.

Baseline Method 1 (C&J'08)

The first baseline method is one proposed by [Chambers and Jurafsky, 2008]. To abstract raw sentences, the heads of predicate verbs and positions of protagonists are focused on (Sec. 1). The model $g(\mathbf{x}, \mathbf{y})$ uses PMI under $\mathcal{Z}$:

$$\mathrm{PMI}(\mathbf{x}, \mathbf{y}; \mathcal{Z}) = \log \frac{N \cdot c(\mathbf{x}, \mathbf{y})}{\sum_{\mathbf{y}'} c(\mathbf{x}, \mathbf{y}') \sum_{\mathbf{x}'} c(\mathbf{x}', \mathbf{y})}, \quad (14)$$

where $c(\mathbf{x}, \mathbf{y})$ denotes the frequency of $(\mathbf{x}, \mathbf{y})$ in $\mathcal{Z}$.

Baseline Method 2 (Jans *et al.*'12)

The second baseline method is proposed by [Jans *et al.*, 2012]. In this method, the abstract representations are identical to those of C&J'08. The model $g(\mathbf{x}, \mathbf{y})$ computes the logarithm of a conditional probability under $\mathcal{Z}$:

$$g(\mathbf{x}, \mathbf{y}) = \log p(\mathbf{y}|\mathbf{x}; \mathcal{Z}) = \log \frac{c(\mathbf{x}, \mathbf{y})}{\sum_{\mathbf{y}'} c(\mathbf{x}, \mathbf{y}')}. \quad (15)$$

Baseline Method 3 (C&J'08 + PHSIC)

We also consider a kernelized version of C&J'08 using PHSIC, which intuitively is *smoothed* PMI. We define the following kernel function between verb dependency tuples (v, d):

$$k((v_1, d_1), (v_2, d_2)) = \begin{cases} \cos(\mathbf{v}(v_1), \mathbf{v}(v_2)) & (d_1 = d_2) \\ -1 & (\text{o.w.}) \end{cases}. \quad (16)$$

Proposed Method (DCSE + PHSIC)

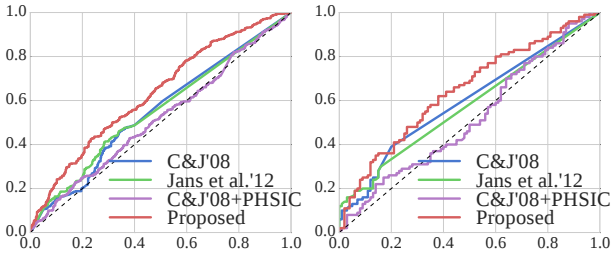
The proposed method uses DCSE, realized using the HSIC and MH for abstraction (Sec. 3 and Sec. 4), and the PHSIC for the model. Note that DCSE is performed for each instance in the test set.

Training

1. Given the training set $\mathcal{D}^{(\mathrm{train})}$, perform DCSE by maximizing the HSIC and generating abstract event pairs $\mathcal{Z}$ (Sec. 2). We ran the MH sampler with $\beta = 10^8$ to draw 7×10^5 and 2×10^5 samples, respectively, for the Gigaword corpus the Fairy Tale corpora.
2. The model is PHSIC under $\mathcal{Z}$ (Eq. 5).

Test

1. Taking $\{(\mathbf{s}, \mathbf{t})\} \cup \mathcal{Z}$ as the input, perform DCSE with a fixed $\mathcal{Z}$ to obtain $(\mathbf{x}, \mathbf{y})$ such that $\mathbf{x} \preccurlyeq \mathbf{s}$, $\mathbf{y} \preccurlyeq \mathbf{t}$.
2. $f(\mathbf{s}, \mathbf{t})$ is defined by $g(\mathbf{x}, \mathbf{y}) = \mathrm{PHSIC}(\mathbf{x}, \mathbf{y}; \mathcal{Z})$.



(a) Gigaword (b) Fairy Tale

Figure 5: ROC curves on the prediction task.

Table 3: AUC values for the prediction task. The best result in each column is shown in bold.

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

Results and Discussion

Fig. 5 shows the ROC curves obtained for each method. The x- and y-axes denote the false positive rate and true positive rate, respectively. The area under the curve corresponds to the AUC. Table 3 summarizes the AUC values of the different methods for each dataset.

The experimental results show that the proposed method outperforms all the existing methods when applied to both the datasets. Moreover, its prediction performance was better than those of the baseline methods over the entire ROC curve (Fig. 5). These results indicate that changing from a fixed abstract representation (C&J) to DCSE resulted in considerable performance improvement in the prediction task.

A comparison between C&J and C&J+PHSIC highlights that there is no advantage of integrating PHSIC with the fixed abstract representation. The experimental results imply that simply applying PHSIC to a fixed abstract representation does not improve predictive performance (C&J'08 + PMI $>$ C&J'08 + PHSIC). These facts also support the effectiveness of determining an abstract representation optimized for each instance (DCSE).

The experimental results for real corpora also show that the proposed method can capture significant association from original sentences more accurately than the existing methods. For instance, from the sentence pair ⟨*Hasegawa had a team-high 10 wins last season.*, *He pitched in with nine saves while Troy Percival was hurt and had an ERA of 3.57 in a team-leading 66 appearances.*⟩ our method extracted ⟨*had wins last season*, *pitched nine saves*⟩, while the existing methods extracted the abstract representation ⟨*X have*, *X pitch*⟩, which cannot readily interpreted.

6 Conclusion

In this paper, we have addressed the problem of determining abstract representations when modeling the associations between items in a dataset. We have proposed a new machine learning task called *dependent co-substructure extraction (DCSE)* that extracts strongly dependent substructure pairs from associated pairs. The proposed method incorporates HSIC (Sec. 3) and MH sampling (Sec. 4) in order to cope with the challenges of data sparsity and huge search space, respectively. Our experimental results demonstrate the effectiveness of the new task and the proposed method in two scenarios, namely knowledge acquisition and predicting narrative event pairs (Sec. 5).

While we obtained favorable experimental results by using a simple cosine kernel, the proposed method can utilize arbitrary kernel functions such as the RBF kernel, tree kernels, and graph kernels on arbitrary data structures such as sequences, graphs, and vectors. An intriguing direction for future work would be to adopt other data structures and kernel functions so that semantic similarities can be captured more precisely. Applying our method to various knowledge mining tasks would also be interesting.

Even with the relatively low computational cost of kernel-based measures, HSIC still faces a scalability problem. Although we conducted experiments on a dataset consisting of 10,000 pairs, we would like to train a better model on a larger dataset with, for example, more than a million pairs. An important task for future work, therefore, is to improve the scalability of proposed method. Promising approaches toward this aim include using various methods for approximating Gram matrices, such as using random Fourier features.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR1513, Japan. We are grateful to Prof. K. Fukumizu and Prof. H. Kashima for giving us valuable advice. We would also like to thank Dr. R. Tian and S. Kobayashi for meaningful discussions.

References

[Chambers and Jurafsky, 2008] Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Event Chains. In *ACL*, pages 789–797, 2008.

[Chambers and Jurafsky, 2009] Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Schemas and their Participants. In *ACL*, pages 602–610, 2009.

[Chib and Greenberg, 1995] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, 1995.

[Church and Hanks, 1990] Kenneth Ward Church and Patrick Hanks. Word Association Norms, Mutual Information, and Lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[Fawcett, 2006] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[Fukumizu *et al.*, 2009] Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Kernel dimension reduction in regression. *Annals of Statistics*, 37(4):1871–1905, 2009.

[Graff and Cieri, 2003] David Graff and Christopher Cieri. English Gigaword, LDC2003T05. *Philadelphia: Linguistic Data Consortium*, 2003.

[Granroth-Wilding and Clark, 2016] Mark Granroth-Wilding and Stephen Clark. What Happens Next? Event Prediction Using a Compositional Neural Network Model. In *AAAI*, pages 2727–2733, 2016.

[Gretton *et al.*, 2005] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring Statistical Dependence with Hilbert-Schmidt Norms. In *ALT*, pages 63–77, 2005.

[Jans *et al.*, 2012] Bram Jans, Steven Bethard, Ivan Vulić, and M. Francine Moens. Skip N-grams and Ranking Functions for Predicting Script Events. In *EACL*, pages 336–344, 2012.

[Maes *et al.*, 1997] Frederik Maes, Andre Collignon, Dirk Vandermeulen, Guy Marchal, and Paul Suetens. Multimodality Image Registration by Maximization of Mutual Information. *IEEE Trans. on Medical Imaging*, 16(2):187–198, 1997.

[Manning *et al.*, 2014] Christopher D. Manning, John Bauer, Jenny Finkel, Steven J. Bethard, Mihai Surdeanu, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL System Demonstrations*, pages 55–60, 2014.

[Muandet *et al.*, 2016] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel Mean Embedding of Distributions: A Review and Beyonds. *arXiv preprint arXiv:1605.09522*, 2016.

[Peng *et al.*, 2005] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[Quadrianto *et al.*, 2009] Novi Quadrianto, Le Song, and Alex J. Smola. Kernelized sorting. In *NIPS*, pages 1289–1296, 2009.

[Resnik, 1997] Philip Stuart Resnik. Selectional Preference and Sense Disambiguation. In *ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How*, pages 52–57, 1997.

[Song *et al.*, 2012] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature Selection via Dependence Maximization. *Journal of Machine Learning Research*, 13:1393–1434, 2012.

[Torkkola, 2003] Kari Torkkola. Feature Extraction by Non-Parametric Mutual Information Maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

[Turney, 2002] Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *ACL*, pages 417–424, 2002.