Modeling Context-sensitive Selectional Preference with Distributed Representations

Naoya Inoue Yuichiro Matsubayashi Masayuki Ono* Naoaki Okazaki Kentaro Inui
Graduate School of Information Sciences
Tohoku University
6-6-05 Aramaki Aza Aoba, Aobaku, Sendai, Miyagi 980-8579, Japan
{naoya-i,y-matsu,masayuki.ono,okazaki,inui}@ecei.tohoku.ac.jp

Abstract

This paper proposes a novel problem setting of selectional preference (SP) between a predicate and its arguments, called as *context-sensitive* SP (CSP). CSP models the narrative consistency between the predicate and preceding contexts of its arguments, in addition to the conventional SP based on semantic types. Furthermore, we present a novel CSP model that extends the neural SP model (Van de Cruys, 2014) to incorporate contextual information into the distributed representations of arguments. Experimental results demonstrate that the proposed CSP model successfully learns CSP and outperforms the conventional SP model in coreference cluster ranking.

1 Introduction

Selectional Preference (SP) of predicates is a term denoting a bias in co-occurrence of a predicate and its argument. Predicates tend to take a particular semantic type of phrase as an argument. For example, the object slot of *eat* is generally filled by a noun phrase denoting food such as *an apple*; it is rarely filled by a phrase that is not food such as *a watch*. As the knowledge of SP has been recognized as key for many natural language processing tasks, including semantic role labeling and anaphora resolution, automatic acquisition of SP knowledge has persisted as a popular research topic. In literature, a variety of computational models for SP have been proposed, ranging from thesaurus-based approaches (Resnik, 1996), to probabilistic latent variable models (Rooth et al., 1999; Séaghdha and Korhonen, 2014), and distributed approaches (Van de Cruys, 2014).

Conventionally, SP is defined as the context-independent acceptability of a word as a filler of a predicate *in the sense of a semantic type*. Suppose that we must identify the referent of $him_{(j)}$:

(1) $John_{(i)}$ *beat* $Bob_{(j)}$. *Mary comforted* $him_{(j)}$.

Henceforth, we call a predicate (e.g., *comfort*) and an argument (e.g., *John* and *Bob*) to be examined as a *query predicate* and *query argument*, respectively. Conventional SP models judge the appropriateness of $John_{(i)}$ and $Bob_{(j)}$ in terms of whether *comfort* can take each noun as its object. However, it ignores the information signified by the preceding context, namely $John_{(i)}$ *beat Bob* and $Bob_{(j)}$, *whom John beat*. Therefore, conventional approaches cannot determine the preference between *John* and *Bob*, both of whom can fill the object of *comfort*, with the same semantic type.

In this paper, we propose a *context-sensitive* version of SP (CSP), a novel task setting in which SP is considered in discourse. In text (1), for instance, $Bob_{(j)}$ is considered to be a more plausible filler than $John_{(i)}$ in terms of contextual relatedness: *one who was beaten* is more likely to be *comforted* than *one who beat someone*. In this paper, we want to discriminate (2a) from (2b) in addition to the conventional SP-based discrimination (e.g., *Mary comforted John* versus *Mary comforted a banana*):

(2) a. *Mary comforted John, who beat Bob.*
b. *Mary comforted Bob, whom John beat.*

*Present affiliation: FUJITSU FIP CORPORATION

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2829–2838, Osaka, Japan, December 11-17 2016.

The goal of this paper is to develop a CSP model that jointly considers the following aspects: (i) the conventional acceptability based on the semantic type of a query argument, and (ii) the narrative consistency between events denoted by a query predicate and preceding context of the query argument (as seen in (3a) with its counter example (3b)).

(3) a. *Mary comforted X who beat Bob.*

b. *Mary comforted X whom John beat.*

The joint modeling has an advantage in applications, such as predicate argument structure analysis and coreference resolution because the preceding context of a given query argument may not always be available in these tasks. For example, in pronoun resolution, some candidate antecedents may have preceding contexts relevant to narrative consistency but other candidates may not.

The challenges in modeling a CSP are as follows: (i) data sparseness caused by the incorporation of context words, and (ii) an effective means of incorporating context-sensitivity into SP. To address these issues, we propose to extend the state-of-the-art SP model by using a distributed representation (Van de Cruys, 2014). The distributed framework alleviates the data sparseness problem and naturally injects the contextual information of a query argument into its word vector based on compositional distributional semantics (Socher et al., 2012; Socher et al., 2013; Muraoka et al., 2014; Hashimoto et al., 2014, etc.).

We empirically evaluate the impacts of incorporating context-sensitivity into SP for two tasks: (i) context-sensitive pseudo-disambiguation, a novel benchmark tailored for evaluating CSP models, and (ii) coreference cluster ranking for pronominal anaphora resolution. The results demonstrate that our approach achieves considerable improvements. Moreover, the results suggest that CSP is a meaningful problem setting and that our model captures the context-sensitivity of SP.

2 Related Work

A fundamental approach to modeling SP is to count the co-occurrences of predicates and their arguments on a large corpus. As simply counting a predicate-argument pair causes data sparseness problem, previous SP models adopted methods for smoothing co-occurrence counts. Earlier efforts combined a manually crafted thesaurus with the acquired distribution (Resnik, 1996; Li and Abe, 1998). Another approach used a latent probabilistic model to obtain a semantically smoothed probability distribution (Rooth et al., 1999; Séaghdha and Korhonen, 2014). Other directions include example-based approaches (Erk, 2007). However, these studies differ from ours in that they do not consider the context-sensitivity.

Some previous studies (Ritter et al., 2010; Van de Cruys, 2014; Kawahara et al., 2014) estimate the plausibility of a subject–verb–object (SVO) tuple. These studies model a type of CSP: a subject or an object can be regarded as an additional context to restrict a set of possible fillers of a query predicate. However, the context captured in our study is not a local context of a query *predicate* but that of a query *argument*, working as a *validator* of the narrative consistency between a query predicate and events in which a query argument participates (see Section 4).

In addition, the modeling of a narrative consistency between events has been studied extensively (Chambers and Jurafsky, 2009; Modi and Titov, 2014; Granroth-wilding and Clark, 2016, etc.). Chambers and Jurafsky (2009) acquired sets of narratively related events sharing at least one entity (e.g., $\{X$ *commit a crime, police arrest* X*,* X *convict, ...*$\}$) by collecting a set of verbal mentions sharing coreferring arguments in a large corpus. The relatedness between two events was then estimated statistically through pointwise mutual information (Church and Hanks, 1990).

To address the data sparseness problem of Chambers and Jurafsky (2009), Granroth-wilding and Clark (2016) proposed an architecture based on distributed representation to judge the narrative coherence between two events. They trained a neural network (NN) model based on event chain instances acquired in the same strategy as that of Chambers and Jurafsky (2009) and reported that the NN model outperformed their approach. As argued in Section 4, our approach can be regarded as an integrated framework of the state-of-the-art approaches of conventional SP and narrative consistency.

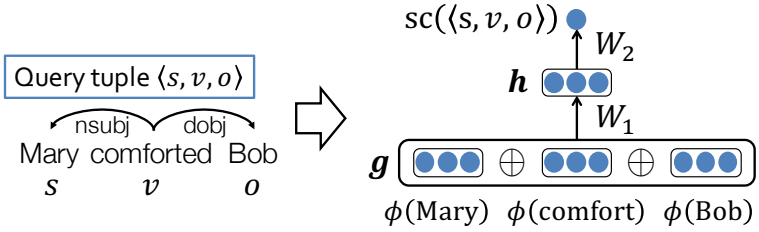

Figure 1: Van de Cruys' SVO model

3 Van de Cruys' SVO model

We adopted the SVO model by Van de Cruys (2014) as a baseline model and extended it to capture narrative consistency. Van de Cruys' SVO model, based on an NN architecture, estimates a preference score for a tuple of words $\langle s, v, o\rangle$ (referred to as a *query tuple*), in which s and o respectively correspond to the subject and object of a transitive verb v. Figure 1 presents the NN structure. The SP score $\mathrm{sc}(\cdot)$ of $\langle s, v, o\rangle$ is calculated using a two-layer NN:

$$\mathrm{sc}(\langle s, v, o\rangle) = W_2 \boldsymbol{h}_{s,v,o}, \tag{1}$$

$$\boldsymbol{h}_{s,v,o} = f(W_1 \boldsymbol{g}_{s,v,o} + \boldsymbol{b}), \tag{2}$$

$$\boldsymbol{g}_{s,v,o} = \phi(s) \oplus \phi(v) \oplus \phi(o) \tag{3}$$

where $\phi(w) \in \mathbb{R}^d$ is the vector representation[1] of word w, $\boldsymbol{g} \in \mathbb{R}^{3d}$ presents an input layer concatenating word vectors of $\langle s, v, o\rangle$ by using the operator $\oplus$, and $\boldsymbol{h} \in \mathbb{R}^h$ is a hidden layer. $W_1 \in \mathbb{R}^{h \times 3d}$ and $W_2 \in \mathbb{R}^{1 \times h}$ are respectively the weight matrices of the first and second layers, and $\boldsymbol{b} \in \mathbb{R}^h$ is a bias on the first layer. $f(\cdot)$ is an element-wise activation function using $\tanh$.

The model simultaneously learns word embedding and scoring function of SP based on the framework proposed by Collobert et al. (2011), who employed a ranking-type loss function that discriminates between positive and negative examples. Positive training examples include $\langle s, v, o\rangle$ tuples observed in a corpus. Negative examples are generated from the positive examples by replacing arguments in correct tuples with randomly selected words. This procedure generates the following three types of negative examples from a positive example $\langle s, v, o\rangle$: $\langle \tilde{s}, v, o\rangle$, $\langle s, v, \tilde{o}\rangle$, and $\langle \tilde{s}, v, \tilde{o}\rangle$. The loss function is then defined as:

$$\sum_{(s,v,o)} \big\{ \max(0, 1 - \mathrm{sc}(\langle s, v, o\rangle) + \mathrm{sc}(\langle \tilde{s}, v, o\rangle)) + \max(0, 1 - \mathrm{sc}(\langle s, v, o\rangle) + \mathrm{sc}(\langle s, v, \tilde{o}\rangle)) \\ + \max(0, 1 - \mathrm{sc}(\langle s, v, o\rangle) + \mathrm{sc}(\langle \tilde{s}, v, \tilde{o}\rangle)) \big\}. \tag{4}$$

Following Collobert et al. (2011), Van de Cruys (2014) calculates the gradient of the loss *online* by sampling a single corrupted subject $\tilde{s}$ and object $\tilde{o}$ for each correct tuple.

4 Context-sensitive SP Model

We propose a context-sensitive selectional preference (CSP) model by extending the SVO model. The advantage of using the model by Van de Cruys (2014) is that we can naturally represent the attachment of the contextual information into a query argument with compositional distribution semantics (Socher et al., 2012; Hashimoto et al., 2014, etc.). We incorporate both the conventional SP and narrative consistency described in Section 1 into a single model by learning the vector representation of *an argument with its context* in the same vector space as word vectors in the SVO model. To realize this, we inserted an additional layer for calculating a context-injected word vector under the input layer.

Figure 2 presents the network structure of our model with the following text as an input.

[1]The original paper differentiates vectors for a word w depending on whether it is used as a subject, object, or verb. However, we used the same vector for a word because we obtained higher performance than that setting.

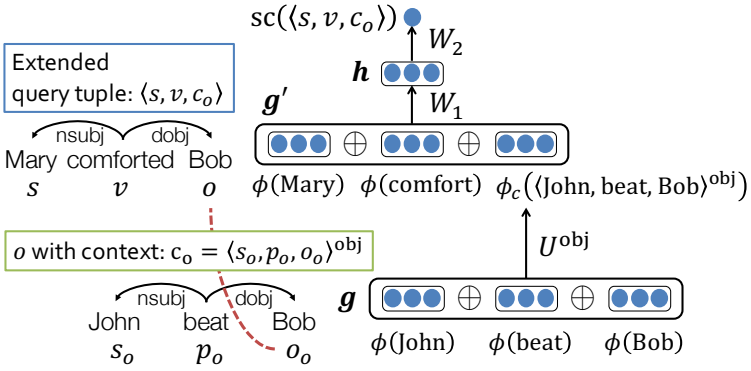

Figure 2: Network structure of proposed model

(4) *John beat Bob*$_{(i)}$. *Mary comforted Bob*$_{(i)}$.

Here, the query tuple $\langle s, v, o\rangle$ is $\langle$*Mary, comfort, Bob*$\rangle$ and the context for the query argument *Bob*$_{(i)}$ is "*John beat Bob*." From this context, we calculate a context-injected word vector representing "*Bob, whom John beat*" by combining the vectors of the words in its predicate-argument structure (PAS). We use the resulting vector as the input to the SVO model, instead of the vanilla word vector.

Formally, we extend the representation of query tuple $\langle s, v, o\rangle$ so that s and o can accompany their contexts. We represent such cases as $\langle c_s, v, o\rangle$, $\langle s, v, c_o\rangle$, and $\langle c_s, v, c_o\rangle$, where c_w denotes *a word w with its context*. Then, we extend Equations (1), (2), and (3) for $\langle s, v, c_o\rangle$ as

$$\mathrm{sc}(\langle s, v, c_o\rangle) = W_2 \boldsymbol{h}'_{s,v,c_o}, \tag{5}$$

$$\boldsymbol{h}'_{s,v,c_o} = f(W_1 \boldsymbol{g}'_{s,v,c_o} + \boldsymbol{b}), \tag{6}$$

$$\boldsymbol{g}'_{s,v,c_o} = f(\phi(s) \oplus \phi(v) \oplus \phi_c(c_o)), \tag{7}$$

where $\phi_c(c_w)$ is a context-injected word vector, which is explained in the rest of this section. Similarly, for $\langle c_s, v, o\rangle$ and $\langle c_s, v, c_o\rangle$, we extend Equation (3) as follows: $\boldsymbol{g}'_{c_s,v,o} = f(\phi_c(c_s) \oplus \phi(v) \oplus \phi(o))$, and $\boldsymbol{g}'_{c_s,v,c_o} = f(\phi_c(c_s) \oplus \phi(v) \oplus \phi_c(c_o))$.

As a context of w, we can potentially consider various types of modifiers, such as predicates, adverbs, appositives, and genitives, that affects the preference score of a query argument. In this study, as a first step, we restrict the context information to PASs along the lines of the previous studies that utilize event-to-event relations in an anaphora resolution (Inoue et al., 2012; Peng et al., 2015).

Specifically, we first assume a context of a query argument be a single PAS which takes the query argument as its argument (referred to as a *context-PAS*). Then we represent w *with its context-PAS* as $c_w = \langle s_w, p_w, o_w\rangle^r$, where s_w and o_w are respectively the subject and object of a predicate p_w that syntactically governs the word w (i.e., either s_w or o_w is w). r indicates the grammatical role of the query argument w in the context-PAS, whose value is either subj (when $w = s_w$) or obj (when $w = o_w$). For example, for text (4), the context for the query object *Bob*, modified by the PAS of the transitive verb *beat*, is represented as $c_o = \langle$*John, beat, Bob*$\rangle^{\mathrm{obj}}$. The grammatical role obj of the query argument *Bob* is indicated to discriminate "*Bob, whom John beat*" from "*John, who beat Bob*." Note that some context-PASs (e.g., intransitive verbs and adjectives) do not take an object argument. r and o_w for these ASs are thus ignored.

To compute a context-injected vector of a query argument, we composed a word vector in a context-PAS by using methods similar to those for building phrase vectors (Socher et al., 2012; Socher et al., 2013; Muraoka et al., 2014; Hashimoto et al., 2014). In this study, we adopted a simple compositional operation, keeping comparisons with other composition methods for future studies. We compute the context-injected word vector $\phi_c(c_w)$ as

$$\phi_c(\langle s_w, p_w, o_w\rangle^r) = U^r \boldsymbol{g}_{s_w,p_w,o_w}, \tag{8}$$

$$\boldsymbol{g}_{s_w,p_w,o_w} = f(\phi(s_w) \oplus \phi(p_w) \oplus \phi(o_w)), \tag{9}$$

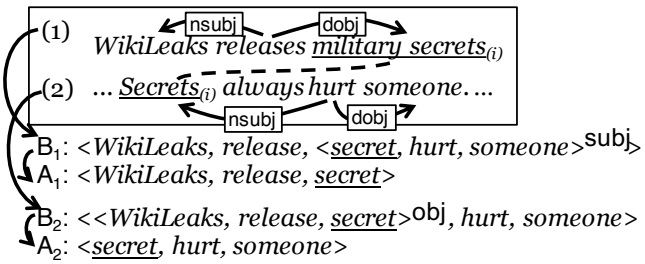

Figure 3: Generation of training instances.

where $U^r \in \mathbb{R}^{h \times 3d}$ is a weight matrix used for building the context-injected vector for the grammatical role r. We set a word vector $\phi(o_w) = \mathbf{0} \in \mathbb{R}^d$ if o_w does not exist (when the context-PAS does not have o_w).

5 Training

To model conventional SP and CSP jointly, we simultaneously train matrices U^r, W_1, and W_2, and vectors $\phi(\cdot)$. We minimize the same loss function introduced in Section 3, except that (i) we replace the score function $\mathrm{sc}(\cdot)$ with Equation (5); and (ii) we use two types of tuples (TYPE A and TYPE B) as training instances. TYPE A is a tuple whose subject and object are bare nouns (i.e., $\langle s, v, o \rangle$). TYPE B is a tuple with either its subject or object including a context-PAS (i.e., $\langle c_s, v, o \rangle$ and $\langle s, v, c_o \rangle$). Hereafter, we describe the method to obtain these instances from a corpus.

5.1 TYPE B instance generation

Positive instance. We assume that a corpus is parsed using a syntactic dependency parser and a coreference resolver. We extract a collection of TYPE B positive instances from the dependency and coreference relations, where we include only a head word for each predicate/argument slot. Figure 3 illustrates the extraction procedure. From sentence (1), we obtain $\langle$*WikiLeaks, release,* $\langle$*secret, hurt, someone*$\rangle^{\mathrm{subj}}\rangle$ (B_1), where "*hurt someone*", the context of *secret*, is attached via the coreference link between *military secrets* and *Secrets*. Similarly, from sentence (2), we obtain $\langle\langle$*WikiLeaks, release, secret*$\rangle^{\mathrm{obj}}$*, hurt, someone*$\rangle$ (B_2).

As context-PASs, we used a *non-negated* transitive verb, intransitive verb, adjective, and copula. Thus, we do *not* extract tuples including one or more negations (e.g., $\langle$John, *not* eat, apple$\rangle$). A nontrivial issue of managing negations in compositional semantics has not been explored much in distributional compositional semantics. Furthermore, we removed tuples where the predicates of coreferent mentions are connected via an *adversative* connective (e.g., *John beat Bob but Bob was happy*), which cannot be handled by the CSP model.

Negative instance. Next, negative training instances are generated by considering positive instances as counterparts. We generate $\langle \tilde{c_s}, v, o \rangle$, $\langle c_s, v, \tilde{o} \rangle$, and $\langle \tilde{c_s}, v, \tilde{o} \rangle$ for positive instance $\langle c_s, v, o \rangle$ and $\langle \tilde{s}, v, c_o \rangle$, $\langle s, v, \tilde{c_o} \rangle$, and $\langle \tilde{s}, v, \tilde{c_o} \rangle$ for positive instance $\langle s, v, c_o \rangle$. Here, $\tilde{s}$ and $\tilde{o}$ are sampled from all the bare subjects and objects in the positive instances, respectively. In addition, $\tilde{c_s}$ and $\tilde{c_o}$ are sampled from all the subjects and objects respectively, with contexts attached. For example, we generate $\tilde{c_s}$=$\langle$*John, eat, apple*$\rangle^{\mathrm{obj}}$ from c_s=$\langle$*WikiLeaks, release, secret*$\rangle^{\mathrm{obj}}$. We use the probabilistic negative sampling (Mikolov et al., 2013) based on the frequency of arguments in the positive instances[2].

5.2 TYPE A instance generation

We then created a TYPE A positive instance from each TYPE B positive instance by replacing the context-attached argument with the original bare argument. In Figure 3, we obtain $\langle$*WikiLeaks, release, secret*$\rangle$ and $\langle$*secret, hurt, someone*$\rangle$ from B_1 and B_2, respectively. To generate TYPE A negative instance, we follow the same procedure described in Section 3 but use probabilistic negative sampling instead.

[2]Although Van de Cruys (2014) used random sampling to generate negative instances, we used probabilistic sampling because our preliminary experiment shows a better performance.

5.3 Dataset

We identified syntactic dependency relations and coreference relations in 4.5 billion sentences extracted from the ClueWeb12 corpus[3], that is, a large collection of Web documents, by applying Stanford CoreNLP (Manning et al., 2014). To reduce noises from the obtained coreference relations, we applied the following heuristics to skim only highly plausible coreference relations off from the pool: (i) the coreference relation must be *intrasentential*, (ii) the head words of the coreferent mentions must be *identical* and *non*pronominal (e.g. *John–John* but not *John–boy*)[4]. Furthermore, we discarded TYPE A and TYPE B instances containing low-frequency words so that all our training instances include only the top 50k frequent verbs, 50k frequent nouns, and 50k frequent adjectives. We replaced all the rare words (occurring less than four times) with the special symbol `OOV` (implying out of vocabulary) to facilitate the calculation of the SP of unseen words appearing in the test set.

As a result, we obtained a collection of 4,824,394 TYPE B positive instances (2,912,624 unique tuples; $\mathcal{B}$ hereafter) and 4,824,394 TYPE A positive instances (1,500,990 unique tuples; $\mathcal{A}$ hereafter).

6 Evaluation

To check whether the CSP model can properly learn the conventional SP and narrative consistency, we first evaluated the CSP model against Van de Cruys' model by using a pseudo-disambiguation test, a binary classification task of discriminating a positive SVO tuple from its pseudo-negative counterpart. We then evaluated the effectiveness of the CSP model in a realistic problem setting, in which the disambiguation test is created from coreference annotations of the OntoNotes corpus (Hovy et al., 2006).

6.1 Parameters

We set the dimension of word embedding $d = 50$ and the dimension of hidden layer $h = 50$. The word embeddings are initialized with the publicly available word vectors trained through GloVe (Pennington et al., 2014)[5] and updated through back propagation. We updated weights by using Adam (Kingma and Ba, 2014) with a mini-batch size of 1,000 and 30 epochs[6].

To evaluate the effectiveness of the CSP model, we replicated the SVO model of Van de Cruys (2014) by training the CSP model only with TYPE A instances (henceforth, SP).

6.2 Pseudo-disambiguation test

Inspired by the conventional SP model (Erk, 2007; Van de Cruys, 2014, etc.), we set up three binary classification tasks. We performed hold-out validation on datasets $\mathcal{A}$ and $\mathcal{B}$.

6.2.1 Tasks

Pseudo-disambiguation (PD) discriminates a positive *non*-context-injected tuple (e.g., $\langle$*Mary, eat, banana*$\rangle$) from its pseudo-negative counterpart ($\langle$*Mary, eat, watch*$\rangle$) without any context information. This task setting has been employed in the previous SP models, including in Van de Cruys (2014).

Context-sensitive PD (CSPD) discriminates a positive context-attached tuple (e.g., $\langle$*Mary, eat,* $\langle$*banana, delicious*$\rangle^{\mathrm{subj}}\rangle$) from its pseudo-negative counterpart ($\langle$*Mary, eat,* $\langle$*watch, new*$\rangle^{\mathrm{subj}}\rangle$). This is a novel task setting designed to highlight the ability of modeling both conventional SP and narrative consistency.

CSPD-X is the same as CSPD except that the coreferent arguments are masked; namely, the task is to discriminate a masked context-attached tuple (e.g. $\langle$*Mary, eat,* $\langle$*X, delicious*$\rangle^{\mathrm{subj}}\rangle$, where *X* denotes the special symbol for masked arguments) from its masked pseudo-negative counterpart. In this task, we set the word vector for the masked argument $\phi(X) = \mathbf{0}$. Forcing the ignorance of the meaning of an argument, this task can assess how precisely the proposed models can predict a query argument through narrative consistency only (i.e., by relying only on the context information).

[3] `http://lemurproject.org/clueweb12/`

[4] A manual inspection revealed that the filtering improved the accuracy of the coreference relations from 71% to 87%.

[5] `http://nlp.stanford.edu/projects/glove/`

[6] All the parameters were determined through our preliminary experiments; we found that all the models were insensitive to the dimension parameters (25, 50, and 100 were explored). An initialization with GloVe performed better than random initialization, and the update of a word vector had a positive impact.

Model	PD	CSPD	CSPD-X
RANDOM	0.5000†	0.5000†	0.5000†
SP	**0.8635**	0.8635	0.5000†
CSP	0.8623	**0.8947***	**0.7856**

Table 1: Accuracy for pseudo-disambiguation tasks. '*' denotes a statistical significance against SP (McNemar test, $p < .05$). '†' indicates the accuracy on random guesses (no clue for discrimination).

Model	MQ	MQ_{no_pr}
SP	0.7420	0.7125
CSP	**0.8265***	0.7586*

Table 2: Model performance on entity ranking. '*' indicates statistical significance against SP (Wilcoxon signed-rank test, $p < .05$).

6.2.2 Dataset

To perform hold-out validation, we first randomly divided the dataset $\mathcal{B}$ into a training set (90%, $\mathcal{B}_{\text{train}}$) and a test set (10%, $\mathcal{B}_{\text{test}}$)[7]. For the PD task, we extracted $\mathcal{A}_{\text{train}}$ from $\mathcal{B}_{\text{train}}$ and $\mathcal{A}_{\text{test}}$ from $\mathcal{B}_{\text{test}}$ by using the procedure described in Section 5.1. For the CSPD and CSPD-X tasks, we used $\mathcal{B}_{\text{train}}$ and $\mathcal{B}_{\text{test}}$. Note that $\mathcal{A}_{\text{train}}$ includes all the SVO instances included in $\mathcal{B}_{\text{train}}$.

6.2.3 Results

Table 1 reports the accuracy of each model in this task. A subtle performance drop (≤ 0.0013 point) is observed in our CSP compared to SP; this was not statistically significant (the statistical significance test by McNemar (1947) showed $p < .05$). This indicates that our joint modeling does not degrade the ability for modeling a conventional SP. In contrast, the CSP model significantly outperformed SP (McNemar test, $p < .05$) in the CSPD task, capturing the context-sensitivity of SP successfully. The results of CSPD-X imply that our joint modeling can properly learn narrative consistency.

6.3 Ranking coreference clusters

In Section 6.2, we reported the results of a binary classification task in which negative instances were artificially generated. In contrast, this section describes a more realistic task setting: *ranking* coreference clusters in the OntoNotes corpus (Hovy et al., 2006).

6.3.1 Task

Given a target pronoun, our task is to determine the coreference cluster (*entity*) that is the most likely to be coreferent with the pronoun. Let us consider text (5) as an example.

(5) *In* his$_{(i)}$ *40-minute* speech$_{(j)}$, Chen$_{(i)}$ *declared the* determination$_{(k)}$ *of the* people$_{(l)}$ ... *against* Chen$_{(i)}$..., *and* he$_{(?)}$ *made a statement...*

Given a target pronoun *he*$_{(?)}$, four coreference clusters C_i, C_j, C_k, C_l are used as candidates for antecedents. As *he*$_{(?)}$ is a subject of the predicate *made a statement*, $\text{sc}(\langle c_s, v, o\rangle)$ gives the preference of c_s as an antecedent of the pronoun, where $v =$ *made* and $o =$ *statement*. In the example, c_s can be a preceding noun with its context attached (if any) or without its context: $\langle$*Chen*, *declare*, *determination*$\rangle^{\text{subj}}$, $\langle$*Chen*, *declare*, *determination*$\rangle^{\text{obj}}$, *speech*, or *people*. We expect that an SP model prefers the correct antecedent $\langle$*Chen*, *declare*, *determination*$\rangle^{\text{subj}}$ over the others.

We measure the ability of a model for selecting the correct cluster by using a *mean quantile* (MQ) (Guu et al., 2015). Let p be the target pronoun, C_p^+ the correct cluster for the pronoun, and $\mathcal{N}_p$ the set of negative (incorrect) clusters. MQ for the pronoun p is defined as:

$$\text{MQ}(p) = \frac{|\{C^- \in \mathcal{N}_p | \text{sp}(C^-, p) < \text{sp}(C_p^+, p)\}|}{|\mathcal{N}_p|}. \tag{10}$$

Intuitively, $\text{MQ}(p)$ represents the ratio where a correct cluster C_p^+ is preferred to incorrect clusters $C^- \in \mathcal{N}_p$ by the model. In general, a cluster contains multiple mentions; thus, we simply consider the maximal of the scores in a cluster C: for example, $\text{sp}(C, p) = \max_{m \in C} \text{sc}(m, v, o)$.

[7]To ensure that the two subsets are strictly disjoint, we prohibited a test instance from being an inverse of any instance in the training set; more concretely, the instance $\langle$*WikiLeaks, release,* $\langle$*secret, hurt, someone*$\rangle^{\text{subj}}\rangle$ must not exist in the test set when the training set includes its inverse $\langle\langle$*WikiLeaks, release, secret*$\rangle^{\text{obj}}$*, hurt, someone*$\rangle$.

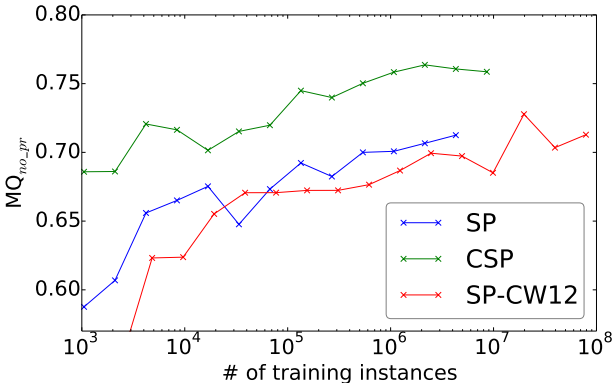

Figure 4: Learning curves of SP and CSP.

When a target pronoun appears at an object position, we use $\mathrm{sc}(\langle s, v, c_o \rangle)$ instead of $\mathrm{sc}(\langle c_s, v, o \rangle)$. In this experiment, we targeted only pronouns filling the subject or object slot of a *non-negated* transitive verb (see Section 5.3); we extracted $he_{(i)}$ but not $his_{(i)}$ in text (5).

6.3.2 Test set

We used the coreference annotations in OntoNotes Corpus 5.0 (Hovy et al., 2006). The corpus includes 625k newswire and 400k broadcast articles annotated with several layers of syntactic and semantic annotations. We obtained 16,414 test pronouns (16.6% of the total pronouns) with the average number of candidate coreference clusters of 75.9. Further, we used semantic roles to extract PASs.

Although pronouns are not informative to SP and CSP, CSP is expected to benefit from context-attached tuples. To test this, we enhanced the training dataset described in Section 5.1 by allowing pronominal coreferent mentions to be extracted because pronouns might appear in coreference clusters. Thus, we obtained a collection of 8,603,782 TYPE B positive instances (4,952,462 unique tuples; $\mathcal{B}_{pro}$ hereafter) and 8,603,782 TYPE A positive instances (2,178,540 unique tuples; $\mathcal{A}_{pro}$ hereafter). We trained SP with $\mathcal{A}_{pro}$, and CSP with $\mathcal{A}_{pro}$ and $\mathcal{B}_{pro}$.

6.3.3 Results

The MQ column of Table 2 shows the mean of the MQ scores for all target pronouns. The proposed model (CSP) outperformed the baseline model (SP) (Van de Cruys, 2014) (Wilcoxon signed-rank test, $p < .05$). The results indicate that the CSP model captures the SPs of predicates more precisely by exploiting the context information of coreference clusters. In addition, our joint models are demonstrated to be capable of comparing query arguments regardless of the existence of context information.

It may be presumed that this improvement is due to the task setting: pronominal coreferent clusters are relatively difficult for SP to discriminate because SP cannot exploit contextual information. Thus, we also report MQ_{no_pr} in Table 2, which is the MQ of the coreference cluster ranking task including only nonpronominal nouns as candidate coreference clusters. This evaluation also shows that our CSP model outperformed the model by Van de Cruys (2014) (Wilcoxon signed-rank test, $p < .05$). The margin in MQ was found to be larger than in MQ_{no_pr}. This indicates that the CSP model successfully captures the narrative consistency-based SP of a pronoun that is difficult to be captured by SP. We leave the effectiveness of context-sensitivity on other (non-neural) types of conventional SP models (e.g., probabilistic latent models (Séaghdha and Korhonen, 2014, etc.)) as an open question. The primary goal of this study is to check the effectiveness of context information; this is proven by the aforementioned results.

6.3.4 Analysis

Although CSP outperformed SP for two tasks, the superiority of CSP may be argued to be due to the larger number of training instances; SP is trained on $\mathcal{A}_{pro}$ but CSP is trained on both $\mathcal{A}_{pro}$ and $\mathcal{B}_{pro}$. Therefore, we plotted the learning curves (`SP` and `CSP` in Figure 4), trained SP and CSP on subsets of

$\mathcal{A}_{pro}$ and $\mathcal{B}_{pro}$ sampled randomly, and measured MQs on the models. As we can generate much a larger size of TYPE A training instance from dependency parses, we also plot the learning curve of the SP model trained using extra SVO tuples extracted from the dependency parse of ClueWeb12 (SP-CW12). We extracted 316,063,648 SVO instances and trained the SP model by using up to 25% of all the extracted SVO tuples (77,011,125 instances) because of the computational cost of training. For fair comparison, we used MQ_{no_pr} as an evaluation measure.

The results show that the MQs of SP (SP, SP-CW12), and CSP (CSP) increase with the size of training data, and both models grow together, keeping a large margin. Based on the growth rate of SP and SP-CW12, we conjectured that we needed 10^3 times more instances of TYPE A for SP to reach the same performance as that of CSP. However, it is inefficient and unrealistic to increase the number of training instances of TYPE A in terms of the training time and availability of training data. In contrast, the CSP model leverages narrative consistency to SP, which is never addressed in previous studies.

To deeply analyze the CSP model, we investigated how the ranks of correct coreference cluster changed from SP to CSP. We found that MQ changed by 0.5 or more in 768 test instances, including 538 improvements and 230 degradations. For 75.7% of the improvements, a context was found to be attached to the candidate antecedent, which is maximally scored among a correct coreference cluster. For example, for the test pronoun *it* in $\langle$*you, own,* $\underline{it}\rangle$, the CSP model can rank the correct antecedent $\langle$*you, buy, something*$\rangle^{\text{obj}}$ at the top by capturing the narrative consistency between *buy X–own X*. In contrast, the context is attached to a correct coreference cluster in only 31.5% of the degradations. This indicates that the CSP model improves the conventional SP via narrative consistency.

7 Conclusion

We addressed the problem of CSP, the novel problem setting of SP. By extending the state-of-the-art SP model (Van de Cruys, 2014), we proposed the novel model that jointly learns both the conventional SP and narrative consistency between a query predicate and its context predicate. The experiments on the PD task demonstrated that the CSP model could leverage narrative consistency for predicting preferences of predicates. Furthermore, the CSP model is effective in a more realistic task setting, ranking coreference clusters.

In the immediate future, we will explore broader contextual information (e.g., prepositional attachment), which can be implemented in our framework naturally. We are interested in applying recent advances in NN, for example, Long Short-Term Memory, Gated Recurrent Unit, and attention mechanism, to compute the vector representation integrating multiple pieces of contextual information. In the experiments, we reserved a downstream application-oriented evaluation for future study so as to exclude various factors specific to a downstream application from the modeling of the CSP. We will explore how to integrate the CSP model with a neural coreference resolver (e.g., Wiseman et al. (2016)) together with conventional coreference features (e.g., distance between a candidate antecedent and a query predicate).

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Numbers 15H01702, 15H05318, 15K16045, and CREST, JST.

References

Nathanael Chambers and Daniel Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL*, pages 602–610.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *CL*, 16(1):22–29.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *ACL*, pages 216–223.

Mark Granroth-wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *AAAI*.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *EMNLP*, pages 318–327.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2014. Jointly learning word representations and composition functions using predicate-argument structures. In *EMNLP*, pages 1544–1555.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *NAACL (short papers)*, pages 57–60.

Naoya Inoue, Ekaterina Ovchinnikova, Kentaro Inui, and Jerry Hobbs. 2012. Coreference resolution with ILP-based weighted abduction. In *COLING*, pages 1291–1308.

Daisuke Kawahara, Daniel W Peterson, and Martha Palmer. 2014. A step-wise usage-based method for inducing polysemy-aware verb classes. In *ACL*, pages 1030–1040.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hang Li and Naoki Abe. 1998. Generalizing case frames using a thesaurus and the mdl principle. *CL*, 24(2):11.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*, pages 55–60.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Ashutosh Modi and Ivan Titov. 2014. Inducing neural models of script knowledge. *CoNLL*, pages 49–57.

Masayasu Muraoka, Sonse Shimaoka, Kazeto Yamamoto, Yotaro Watanabe, Naoaki Okazaki, and Kentaro Inui. 2014. Finding the best model among representative compositional models. In *PACLIC 28*, pages 65–74.

Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving hard coreference problems. In *NAACL*, pages 809–819.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Philip Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61(1):127–159.

Alan Ritter, Oren Etzioni, et al. 2010. A latent dirichlet allocation method for selectional preferences. In *ACL*, pages 424–434.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *ACL*, pages 104–111.

Diarmuid Séaghdha and Anna Korhonen. 2014. Probabilistic distributional semantics with latent variable models. *CL*, 40(3):587–631, September.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP*, pages 1201–1211.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y Ng. 2013. Parsing with compositional vector grammars. In *ACL*, pages 455–465.

Tim Van de Cruys. 2014. A neural network approach to selectional preference acquisition. In *EMNLP*, pages 26–35.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *NAACL*, pages 994–1004.