Neural Gibbs Sampling for Joint Event Argument Extraction

Xiaozhi Wang[1*], **Shengyu Jia**[3*], **Xu Han**[1], **Zhiyuan Liu**[1,2†],
Juanzi Li[1,2], **Peng Li**[4], **Jie Zhou**[4]
[1]Department of Computer Science and Technology, BNRist;
[2]KIRC, Institute for Artificial Intelligence;
[3]Department of Electrical Engineering,
Tsinghua University, Beijing, 100084, China
[4]Pattern Recognition Center, WeChat AI, Tencent Inc, China
{wangxz20, jsy20, hanxu17}@mails.tsinghua.edu.cn

Abstract

Event Argument Extraction (EAE) aims at predicting event argument roles of entities in text, which is a crucial subtask and bottleneck of event extraction. Existing EAE methods either extract each event argument roles independently or sequentially, which cannot adequately model the joint probability distribution among event arguments and their roles. In this paper, we propose a Bayesian model named Neural Gibbs Sampling (NGS) to jointly extract event arguments. Specifically, we train two neural networks to model the prior distribution and conditional distribution over event arguments respectively and then use Gibbs sampling to approximate the joint distribution with the learned distributions. For overcoming the shortcoming of the high complexity of the original Gibbs sampling algorithm, we further apply simulated annealing to efficiently estimate the joint probability distribution over event arguments and make predictions. We conduct experiments on the two widely-used benchmark datasets ACE 2005 and TAC KBP 2016. The Experimental results show that our NGS model can achieve comparable results to existing state-of-the-art EAE methods. The source code can be obtained from https://github.com/THU-KEG/NGS.

1 Introduction

Event argument extraction (EAE) is a crucial subtask of Event Extraction, which aims at predicting entities and their event argument roles in event mentions. For instance, given the sentence "Fox's stock price rises after the acquisition of its entertainment businesses by Disney", the event detection (ED) model will first identify the trigger word "acquisition" triggering a Transfer-Ownership event. Then, with the trigger word and event type,

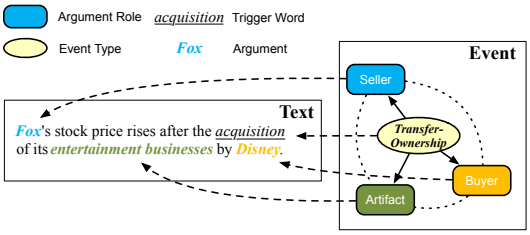

Figure 1: An example of event extraction, including event detection and event argument extraction.

the EAE model is required to identify that "Fox" and "Disney" are event arguments whose roles are "Seller" and "Buyer" respectively. As ED is well-studied in recent years (Liu et al., 2018a; Nguyen and Grishman, 2018; Zhao et al., 2018; Wang et al., 2019a), EAE becomes the bottleneck and has drawn growing attention.

As EAE is the bottleneck of event extraction, especially is also important for various NLP applications (Yang et al., 2003; Basile et al., 2014; Cheng and Erk, 2018), intensive efforts have already been devoted to designing effective EAE systems. The early feature-based methods (Patwardhan and Riloff, 2009; Gupta and Ji, 2009) manually design sophisticated features and heuristic rules to extract event arguments. As the development of neural networks, various neural methods adopt convolutional (Chen et al., 2015) or recurrent (Nguyen et al., 2016) neural networks to automatically represent sentence semantics with low-dimensional vectors, and independently determine argument roles with the vectors. Recently, some advanced techniques have also been adopted to further enhance the performance of EAE models, such as zero-shot learning (Huang et al., 2018), multi-modal integration (Zhang et al., 2017) and weak supervision (Chen et al., 2017).

However, above-mentioned methods do not model the correlation among event arguments in

* indicates equal contribution
† Corresponding author: Z.Liu (liuzy@tsinghua.edu.cn)

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 169–180
December 4 - 7, 2020.

event mentions. As shown in Figure 1, all event arguments are correlated with each other. It is more likely to see a "`Seller`" when you have seen a "`Buyer`" and an "`Artifact`" in event mentions, and vice versa. Formally, with x_i denoting the random variable of the i-th event argument candidate, the required probability distribution for EAE is $P(x_1, x_2, \ldots, x_n|o)$, where o is the observation from sentence semantics of event mentions. The existing methods which independently extract event arguments solely model $P(x_i|o)$, totally ignoring the correlation among event arguments, which may lead models to trapping in a local optimum.

Recently, some proactive works view EAE as a sequence labeling problem (Yang and Mitchell, 2016; Nguyen et al., 2016; Zeng et al., 2018) and adopt conditional random field (CRF) with the Viterbi algorithm (Rabiner, 1989) to solve the problem. These explorations consider the correlation of event arguments unintentionally. Yet limited by the Markov property, their linear-chain CRF only considers the correlation between two adjacent event arguments in the sequence and finds a maximum likelihood path to model the joint distribution, i.e, these sequence models cannot adequately handle the complex situation that each event argument is correlated with each other in event mentions, just like the example shown in Figure 1.

To adequately model the genuine joint distribution $P(x_1, x_2, \ldots, x_n|o)$ rather than $\prod_i^n P(x_i|o)$ for EAE, we propose a Bayesian method named **Neural Gibbs Sampling (NGS)** inspired by previous work (Finkel et al., 2005; Sun et al., 2014). Gibbs sampling (Geman and Geman, 1987) is a Markov Chain Monte Carlo (MCMC) algorithm, which defines a Markov chain in the space of possible variable assignments whose stationary distribution is the desired joint distribution. Then, a Monte Carlo method is adopted to sample a sequence of observations, and the sampled sequence can be used to approximate the joint distribution.

More specifically, for NGS, we first adopt a neural network to model the prior distribution $P_p(x_i|o)$ and independently predict an argument role for each event argument candidate to get an initial state for the random variable sequence $x_1, x_2, \ldots, x_n$, which is similar to the previous methods. Then, we train a special neural network to model the conditional probability distribution $P_c(x_i|x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n, o)$ and iteratively change the sequence state by this conditional distribution. Intuitively, the network modeling the conditional probability distribution aims to predict unknown argument roles based on both sentence semantics and some known argument roles. After enough steps, the state of the sequence will accurately follow the posterior joint distribution $P(x_1, x_2, \ldots, x_n|o)$, and the most frequent state in history will be the best result of EAE.

Considering that it will take many steps to accurately estimate the shape of the joint distribution and each step uses neural networks for inference, it is time-consuming and impractical. Due to what we want for EAE is the max-likelihood state of the argument roles, we follow Geman and Geman (1987) and adopt **simulated annealing** (Kirkpatrick et al., 1983) to efficiently find the max-likelihood state based on the Gibbs sampling.

To conclude, our main contributions can be summarized as follows:

(1) Our NGS method combines both the advantages of neural networks and the Gibbs sampling method. The neural networks have shown their strong ability to fit a distribution from data. Gibbs sampling has remarkable advantages in performing Bayesian inference and modeling the complex correlation among event arguments.

(2) Considering the shortcoming of high complexity of the original Gibbs sampling algorithm, we further apply simulated annealing to efficiently estimate the joint probability distribution and find the max-likelihood state for NGS.

(3) Experimental results on the widely-used benchmark datasets ACE 2005 and TAC KBP 2016 show that our NGS works well to consider the correlation among event arguments and achieves the state-of-the-art results. The experiments also show that the simulated annealing method can significantly improve the convergence speed and the stability of Gibbs sampling, which demonstrate that our NGS is both effective and efficient.

2 Related Work

Event Extraction (EE) aims to extract structured information from plain text, which is a challenging task in the field of information extraction. EE consists of two subtasks, one is event detection (ED) to detect words triggering events and identify event types, the other is event argument extraction (EAE) to extract argument entities in event mentions and identify event argument roles. As EE is important and beneficial for various downstream

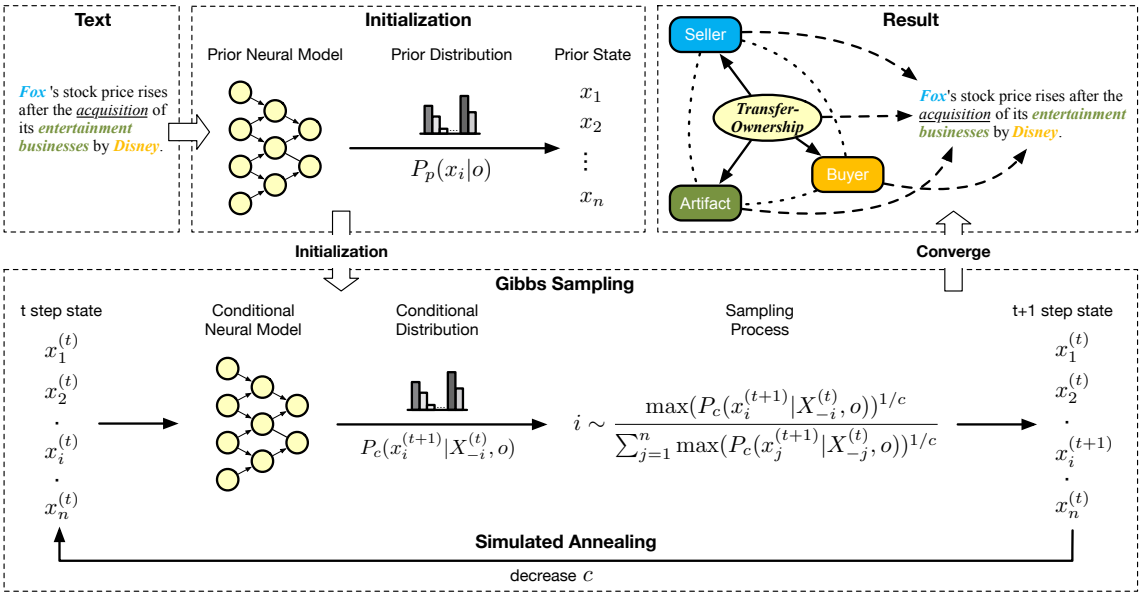

Figure 2: Overall framework of our Neural Gibbs Sampling model.

NLP tasks, e.g., question answering (Yang et al., 2003), information retrieval (Basile et al., 2014), and reading comprehension (Cheng and Erk, 2018), it has attracted wide attentions recently.

ED has been well-studied by the previous works due to its simple and clear definition, including feature-based and rule-based methods (Ahn, 2006; Ji and Grishman, 2008; Gupta and Ji, 2009; Riedel et al., 2010; Hong et al., 2011; McClosky et al., 2011; Huang and Riloff, 2012a,b; Araki and Mitamura, 2015; Li et al., 2013; Yang and Mitchell, 2016; Liu et al., 2016b), neural methods (Chen et al., 2015; Nguyen and Grishman, 2015; Nguyen et al., 2016; Duan et al., 2017; Nguyen et al., 2016; Ghaeini et al., 2016; Lin et al., 2018), the methods with external heterogeneous knowledge (Liu et al., 2016a, 2017; Zhang et al., 2017; Duan et al., 2017; Zhao et al., 2018; Liu et al., 2018b). Some advanced architectures, such as graph convolutional networks (Nguyen and Grishman, 2018) and adversarial training (Hong et al., 2018; Wang et al., 2019a), have also been applied recently.

As ED models has achieved relatively promising results, the more difficult EAE becomes the bottleneck of EE, and have drawn growing research interests. The early works (Patwardhan and Riloff, 2009; Gupta and Ji, 2009; Liao and Grishman, 2010b,a; Huang and Riloff, 2012b; Li et al., 2013) focus on designing hand-crafted features and heuristic rules to extract event arguments, which suffer from the problem of both implementation complexity and low recall. As the rapid development of neural networks, various neural methods have been proposed, such as utilizing convolutional models (Chen et al., 2015), utilizing recurrent models (Nguyen et al., 2016; Sha et al., 2018), and fine-tuning pre-trained language model BERT (Wang et al., 2019b). As compared with the early feature-based and rule-based methods, neural methods automatically represent sentence semantics with low-dimensional vectors, and independently determine argument roles with the vectors, leading to getting rid of designing sophisticated features and rules. Recently, some works adopt some advanced techniques to further improve EAE models in different scenarios, including zero-shot learning (Huang et al., 2018), multi-modal integration (Zhang et al., 2017), cross-lingual (Subburathinam et al., 2019), end-to-end (Wadden et al., 2019), and weak supervision (Chen et al., 2017; Zeng et al., 2018).

The current methods for EAE have achieved some promising results. However, they focus on independently handling each argument entity to predict its role. Because of ignoring to capture rich correlated knowledge among event arguments, the above-mentioned methods are easy to trap in a local optimum and make some inexplicable mistakes. Inspired by some methods in named entity recognition (Huang et al., 2015) and relation extraction (Miwa and Bansal, 2016), some recent proactive works view EAE as a sequence labeling problem. Following the methods for sequence labeling problem (Ma and Hovy, 2016), these sequential EAE models (Yang and Mitchell, 2016; Zeng et al.,

2018) adopt conditional random field (CRF) with the Viterbi algorithm (Rabiner, 1989), and unintentionally consider the correlation of event arguments. Limited by the Markov property, the linear-chain CRF sequentially considers the correlation between two adjacent event arguments, which cannot adequately handle the complex situation in EAE that each argument and any other arguments may be correlated. To this end and inspired by some proactive works (Finkel et al., 2005; Sun et al., 2014), we adapt Gibbs sampling (Geman and Geman, 1987) for EAE to perform approximate inference from the joint distribution. Moreover, we incorporate simulated annealing (Kirkpatrick et al., 1983) to accelerate the sampling process, leading to an effective and efficient method.

3 Methodology

3.1 Framework

For convenience, we denote $X = \{x_1, \ldots, x_n\}$ and $X_{-i} = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$. Figure 2 shows the overall framework of our Neural Gibbs Sampling (NGS) method, consisting of the following modules:

The neural models, including a prior neural model to model the prior distribution $P_p(x_i|o)$, and a conditional neural model to model the conditional distribution $P_c(x_i|X_{-i}, o)$. The prior neural model is similar with existing EAE methods, which takes the event mention text as input and outputs the labels of event argument candidates. The labels will serve as the prior state for the Gibbs sampling module. The conditional neural model takes the text and the results of the last step as input and outputs the probability distribution over labels for each event argument candidate.

The Gibbs sampling module to sample variable assignments X with $P_p(x_i|o)$ and $P_c(x_i|X_{-i}, o)$, which gradually match the implicit posterior joint distribution.

The simulated annealing method to efficiently find the optimal state in the Markov chain of Gibbs sampling. It uses a "temperature" parameter to control the sharpness of the transition distribution. With the "temperature" decreasing, the algorithm will more and more tend to choose the max-likelihood state as the next state.

3.2 Neural Models

The Prior Neural Model is to model the prior distribution $P_p(x_i|o)$. In this paper, we use DMCNN (Chen et al., 2015) and DMBERT as the prior neural models. Given a sentence consisting of several words $\{w_1, \ldots, t, \ldots, w_i, \ldots, w_n\}$, where t and w_i denote the trigger word and the candidate argument entity respectively.

DMCNN transfers each word in the word sequence into an input embedding $\mathbf{e}_i$, which consists of word embedding, event type embedding, and position embedding. Then, DMCNN feeds the input embeddings into a convolutional encoding layer to automatically learn the features and a dynamic multi-pooling layer to aggregate the features into a unified sentence observation embedding to predict an argument role x_i for w_i.

DMBERT is a variation of BERT (Devlin et al., 2019) proposed by Wang et al. (2019b). It adopts a pre-trained BERT to represent the word sequence as feature vectors and also uses a dynamic multi-pooling mechanism like DMCNN to aggregate the features into an instance embedding for prediction. It inserts special tokens around the event argument candidates to indicate their positions.

We sample an argument role following $P_p(x_i|o)$ for each argument candidate and finally predict an initial argument role state $X^{(0)} = \{x_1^{(0)}, \ldots, x_n^{(0)}\}$ as the start point of Gibbs sampling. Note that, our NGS method does not have any special requirements for the prior neural model, any other neural networks can also be used.

Conditional Neural Model is to model the conditional distribution $P_c(x_i|X_{-i}, o)$ for the state transition in Gibbs sampling. Considering that it requires to integrate the argument role information of X_{-i} to compute $P_c(x_i|X_{-i}, o)$, we set an argument role embedding $\mathbf{a}_i$ for each word w_i to represent whether it is an event argument and which role it is of. Then, we modify the input layer of DMCNN and DMBERT to feed the argument role embeddings in. More specifically, DMCNN concatenates the original input embedding $\mathbf{e}_i$ with the argument role embedding $\mathbf{a}_i$ as new inputs. DMBERT utilizes the pre-trained parameters and adds $\mathbf{a}_i$ into the input embedding.

3.3 Gibbs Sampling Module

The Gibbs sampling module aims at sampling from the implicit joint distribution $P(X|o)$. As Algorithm 1 shows, we use the prior neural model to initialize an initial state $X^{(0)}$. In step t, for each random variable x_i, we input the other random variables' states $X_{-i}^{(t-1)}$ into the conditional neu-

Algorithm 1 Neural Gibbs sampling

Input: Initial state $X^{(0)} = \{x_1^{(0)}, \ldots, x_n^{(0)}\}$ predicted by the prior neural network

Result: N samples matching the joint distribution $P(X|o)$

Train the conditional neural model to fit $P_c(x_i|X_{-i}, o)$

for $t \leftarrow 1$ **to** N **do**

  // iteratively change the state

  for $i \leftarrow 1$ **to** n **do**

    $x_i^{(t)} \leftarrow \text{sample}\left(P_c\left(x_i^{(t)}|X_{-i}^{(t-1)}, o\right)\right)$

  end

  $X^{(t)} \leftarrow \{x_1^{(t)}, \ldots, x_n^{(t)}\}$

end

Return $X^{(1)}, \ldots, X^{(N)}$

Algorithm 2 NGS + simulated annealing

Input: Initial state $X^{(0)} = \{x_1^{(0)}, \ldots, x_n^{(0)}\}$ predicted by the prior neural network

Result: The max-likelihood state $X^{(N)}$

Train the conditional neural model to fit $P_c(x_i|X_{-i}, o)$

$c = 1$

for $t \leftarrow 1$ **to** N **do**

  // randomly choose i to transit

  $i \leftarrow \text{sample}\left(\frac{\max\left(P_c\left(x_i^{(t)}|X_{-i}^{(t-1)}, o\right)\right)^{1/c}}{\sum_{j=1}^{n}\max\left(P_c\left(x_j^{(t)}|X_{-j}^{(t-1)}, o\right)\right)^{1/c}}\right)$

  $x_i^{(t)} \leftarrow \arg\max\left(P_c\left(x_i^{(t)}|X_{-i}^{(t-1)}, o\right)\right)$

  $X^{(t)} \leftarrow X_{-i}^{(t-1)} \cup \{x_i^{(t)}\}$

  decrease c

end

Return $X^{(N)}$

ral model to get the distribution $P_c\left(x_i^{(t)}|X_{-i}^{(t-1)}, o\right)$. Then we sample $x_i^{(t)}$ from the distribution, and finally get the new state $X^{(t)}$. We can approximately sample N samples $X^{(1)}, \ldots, X^{(N)}$ with the Gibbs sampling module. Our Appendix gives the proof that the samples will accurately follow the joint distribution after enough steps.

Geman and Geman (1987) have shown that the samples from the beginning of the Markov chain (the burn-in period) may not accurately follow the desired distribution, hence we choose the most frequent state from $X^{(\frac{N}{2})}, \ldots, X^{(N)}$ as the result.

3.4 Simulated Annealing Method

The Gibbs sampling module is to accurately estimate the shape of $P(X|o)$, which will take many steps to reach the convergence. As what we want for EAE is only the max-likelihood state, we adopt a simulated annealing method to efficiently find the optimal state following Geman and Geman (1987).

As shown in Algorithm 2, in step t, the simulated annealing method randomly sample an i from the distribution $\frac{\max\left(P_c\left(x_i^{(t)}|X_{-i}^{(t-1)}, o\right)\right)^{1/c}}{\sum_{j=1}^{n}\max\left(P_c\left(x_j^{(t)}|X_{-j}^{(t-1)}, o\right)\right)^{1/c}}$. The probability of i being chosen has positive correlation with the probability of the max-likelihood state in the conditional distribution of x_i. Then we only need to update x_i with its max-likelihood state in conditional distribution $P_c\left(x_i^{(t)}|X_{-i}^{(t-1)}\right)$ modeled by the conditional neural model to get the next state $X^{(t)}$, which is more efficient than the original Gibbs sampling method. The simulated annealing method adopts a time-varying parameter c to control the sharpness of the distribution. With c gradually decreasing, the algorithm more and more tends to transit in the max-likelihood way and will quickly reach the max-likelihood state. When c is large, it performs like the original Gibbs sampling, so that can avoid falling into suboptimal results.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate the proposed models on two real-world datasets: the most widely-used ACE 2005 (Walker et al., 2006) and the newly-developed TAC KBP 2016 (Ellis et al., 2015). They are both often used as the benchmark in the previous works.

ACE 2005 [1] is the most widely-used dataset in EE, consisting of 599 documents, 8 event types, 33 event subtypes, and 35 argument roles. We evaluate our models by the performance of argument classification. When testing models, an argument is correctly classified only if its event subtype, offsets and argument role match the annotation results. For fair comparison with the previous works (Liao and Grishman, 2010b; Chen et al., 2015), we follow them to use the same test set containing 40 newswire documents, the similar development set with 30 randomly selected documents and training set with the remaining 529 documents.

TAC KBP 2016 [2] indicates the data of the TAC KBP 2016 Event Argument Extraction track, which is the latest benchmark dataset in EE. Different

[1] https://catalog.ldc.upenn.edu/LDC2006T06

[2] https://tac.nist.gov//2016/KBP/

from ACE 2005, this competition only annotates difficult test data but no training data. Accordingly, they encourage participants to construct training data from any other sources by themselves. Considering the argument roles of TAC KBP 2016 are almost the same with ACE 2005 expect TAC KBP 2016 merges all the time-related roles in ACE 2005. We use the ACE 2005 dataset as our training data, which is also provided to the participants of the competition. Hence we can have a fair comparison with the baselines.

For fair comparison with the baselines, we use the same evaluation metrics with previous works: (1) **Precision (P)**, which is defined as the number of correct argument predictions divided by the number of all argument predictions returned by the model. (2) **Recall (R)**, which defined as the number of correct argument predictions divided by the number of all correct golden results in the test set. (3) **F1 score (F1)**, which is defined as the harmonic mean of the precision and recall. F1 score is the most important metric to evaluate EAE performance.

4.2 Baselines

To directly show the improvement of our method from the comparisons, we reproduce **DMCNN** and **DMBERT** as baselines on both of the two datasets. In addition, we also select some state-of-the-art baselines on the two datasets respectively.

On **ACE 2005**, we compare our models with various state-of-the-art baselines, including: (1) Feature-based methods. **Li's joint** (Li et al., 2013) adopts structure prediction to extract events, which is the best traditional feature-based method. **RBPB** (Sha et al., 2016) adopts a regularization-based method to balance the effect of features and patterns, and also consider the relationship between argument candidates. (2) Vanilla neural network methods. **JRNN** (Nguyen et al., 2016) jointly conducts event detection and event argument extraction with bidirectional recurrent neural networks. (3) Advanced neural network method with external information. The **dbRNN** (Sha et al., 2018) utilizes a recurrent neural network with dependency bridges to carry syntactically related information between words, which considers not only sequence structures but also tree structures of the sentences. The **HMEAE** (Wang et al., 2019b) leverages the latent concept hierarchy among argument roles with neural module networks, which considers the label dependency but still classify each event argument independently.

Learning Rate	10^{-3}
Batch Size	60
Dropout Probability	0.5
Hidden Layer Dimension	300
Kernel Size	3
Word Embedding Dimension	100
Position Embedding Dimension	5
Event Type Embedding Dimension	5
Argument Role Embedding Dimension	5

Table 1: Hyperparameter settings for CNN models.

Learning Rate	6×10^{-5}
Batch Size	50
Warmup Rate for the Prior Neural Model	0.1
Warmup Rate for the Conditional Nueral Model	0.05
Argument Role Embedding Dimension	768

Table 2: Hyperparameter settings for BERT models.

On **TAC KBP 2016**, we compare our models with the top systems of the competition, including: **DISCERN-R** (Dubbin et al., 2016), **CMU CS Event1** (Hsi et al., 2016), **Washington1** and **Washington4** (Ferguson et al., 2016).

4.3 Hyperparameter Settings

Our methods with DMCNN and DMBERT as the prior and conditional neural networks are named as **NGS (CNN)** and **NGS (BERT)** respectively. They both transit for 200 steps and the c linearly decrease from 1 to 0. As our work focuses on extracting event arguments and their roles and our methods do not involve the event detection stage (to identify the trigger and determine the event type), we conduct EAE based on the event detection models in (Chen et al., 2015) and (Wang et al., 2019a) for the CNN and BERT models respectively.

For **NGS (CNN)**, the hyperparameters of the prior and conditional neural networks are set as the same as in the original **DMCNN** (Chen et al., 2015). We also use the pre-trained word embeddings learned by Skip-Gram (Mikolov et al., 2013) as the initial word embeddings. The detailed hyperparameters are shown in Table 1.

For **NGS (BERT)**, the two BERT models for the prior and conditional probability distributions are both based on the $\text{BERT}_{\text{BASE}}$ model in Devlin et al. (2019). We apply the pre-trained model [3] to initialize the parameters. To utilize the event type information in our model, we append a special token into each input sequence for BERT to indicate

[3] github.com/google-research/bert

Method	Trigger Classification			Argument Role Classification		
	P	R	F1	P	R	F1
Li's Joint	73.7	62.3	67.5	64.7	44.4	52.7
DMCNN	75.6	63.6	69.1	62.2	46.9	53.5
RBPB	70.3	67.5	68.9	54.1	53.5	53.8
JRNN	66.0	73.0	69.3	54.2	56.7	55.4
HMEAE (CNN)	75.6	63.6	69.1	57.3	54.2	55.7
DMBERT	77.6	71.8	74.6	58.8	55.8	57.2
dbRNN	74.1	69.8	71.9	**66.2**	52.8	58.7
HMEAE (BERT)	77.6	71.8	74.6	62.2	56.6	59.3
NGS (CNN)	75.6	63.6	69.1	61.3	51.3	55.9
NGS (BERT)	77.6	71.8	74.6	59.9	**59.1**	**59.5**

Table 3: The overall EAE results (%) of various baselines and NGS on ACE 2005. EAE performances are influenced by the trigger quality, hence we also provide the trigger classification (event detection) results. Note that as our work does not involve the event detection stage, the NGS (CNN) and NGS (BERT) use the triggers predicted by DMCNN and DMBERT respectively.

Method	Argument Role Classification		
	P	R	F1
DISCERN-R (Dubbin et al., 2016)	7.9	7.4	7.7
Washington4 (Ferguson et al., 2016)	32.1	5.0	8.7
CMU CS Event1 (Hsi et al., 2016)	**31.2**	4.9	8.4
Washington1 (Ferguson et al., 2016)	26.5	6.8	10.8
DMCNN (Chen et al., 2015)	17.9	16.0	16.9
HMEAE (CNN) (Wang et al., 2019b)	15.3	22.5	18.2
DMBERT (Wang et al., 2019b)	22.6	24.7	23.6
HMEAE (BERT) (Wang et al., 2019b)	24.8	**25.4**	25.1
NGS (CNN)	21.5	16.2	18.5
NGS (BERT)	25.5	25.1	**25.3**

Table 4: The overall EAE results (%) of various baseline methods and our NGS on TAC KBP 2016 Event Argument Task. All the models use golden triggers.

the event type. Additional hyperparameters used in our experiments are shown in Table 2.

4.4 Overall Evaluation Results

The overall results of various baseline methods and NGS on ACE 2005 are shown in Table 3. And the results on TAC KBP 2016 are shown in Table 4. From the results, we observe that:

(1) NGS (CNN) and NGS (BERT) achieve significant improvements as compared with DMCNN and DMBERT respectively. Meanwhile, our models still outperform other baseline methods, which are either the typical EAE models or the recent state-of-the-art models. It indicates that our Gibbs sampling with simulated annealing works well to improve EAE with the help of adequately modeling the correlation between event arguments. This demonstrates that our method is effective.

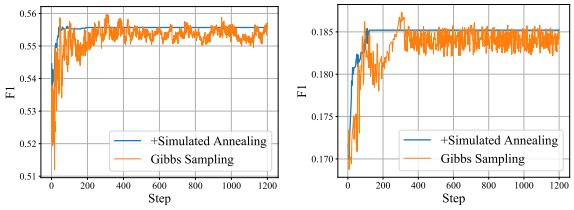

Figure 3: F1-step curves of NGS (CNN) with the simulated annealing method and the original Gibbs sampling on ACE 2005 (left) and TAC KBP 2016 (right).

(2) As NGS enhances both CNN models and BERT models on different datasets, it shows that our Gibbs sampling with simulated annealing is independent of EAE models. In other words, our method can be easily adapted for other EAE models to enhance their extraction performances.

(3) From the experimental results on both ACE 2005 and TAC KBP 2016, we can find that the recall scores and F1 scores of our models are much better than the baseline models. The precision scores of our models do not achieve such obvious improvements. This is consistent with what we mention in the previous sections.

We argue that the baseline models focusing on independently handling each event argument candidates may sever the constraints among argument roles, and may trap in a local optimum or over-fit the training set. The models without considering argument correlations may predict various argument roles with high confidence, even make some inexplicable mistakes. Hence the precision scores of these models may increase, but their recall scores and F1 scores may decrease.

Our models adopt Gibbs sampling for EAE to perform approximate inference from the joint distribution, and make the most of the corrleation and constraints among argument roles. Accordingly, our models can avoid these issues and achieve the state-of-the-art results.

4.5 Ablation Study

In order to verify the effectiveness of our method, especially for the simulated annealing method and the prior neural network, we conduct ablation studies on ACE 2005 and TAC KBP 2016.

Effectiveness of the Simulated Annealing

To demonstrate the effectiveness of the simulated annealing method, we show the F1-step curves of

Type: Justice Subtype: Appeal					
Text: **Malaysia**'s second highest **court** on **Friday** rejected an appeal by ... **Anwar Ibrahim** against his conviction and nine-year prison sentence for **sodomy**.					
Event Argument Candidate	Malaysia	court	Friday	Anwar Ibrahim	sodomy
DMCNN	Place✓	Adjudicator✓	Time-Within✓	Plaintiff✓	N/A×
NGS (CNN)	Place✓	Adjudicator✓	Time-Within✓	Plaintiff✓	Crime✓

Table 5: Top: An example sentence highlighting the event argument candidates, which is sampled from ACE 2005. Bottom: EAE results of DMCNN and NGS (CNN). NGS (CNN) correctly classifies "sodomy" into Crime with the help of correlations among event arguments.

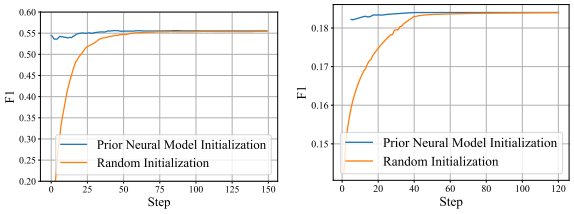

Figure 4: F1-step curves of NGS (CNN) with prior neural network initialization and random initialization on ACE 2005 (left) and TAC KBP 2016 (right).

Gibbs sampling with and without the simulated annealing in Figure 3. We can observe that:

(1) The simulated annealing method can significantly improve the convergence speed and the stability. Our methods just require quarter to half of the steps to reach the convergence.

(2) The simulated annealing method does not weaken the performance of our models. Although the methods with the simulated annealing are much more efficient than those without the simulated annealing, their results are comparable.

Effectiveness of the Prior Neural Network

As the mathematical proof in the Appendix shows, a prior distribution is not necessary for Gibbs sampling. To demonstrate the effectiveness of the prior neural model, we show the F1-step curves of the prior neural model initialization and a random initialization for our NGS method (with simulated annealing) in Figure 4. As it shows in figures, our NGS models with the prior neural network initialization take much fewer steps to reach the convergence than those models with random initialization, which is important and meaningful for the application. Combining the prior neural network initialization and the simulated annealing for our NGS will lead to a more efficient model.

#arguments	1-2	3-4	>5
DMCNN	55.3	54.1	61.8
NGS (CNN)	56.7 (+1.4)	57.9 (+3.8)	69.5 (+7.7)

Table 6: F1 scores (%) of DMCNN and NGS (CNN) on different parts of ACE 2005 dev set with different event argument numbers per sentence.

4.6 Analysis on Modeling Event Argument Correlations

To analyze whether NGS can successfully capture the event argument correlations and further improve EAE performance, we conduct a case study in Table 5 and a quantitative analysis in Table 6.

The sentence in Table 5 is a real sentence containing an Appeal event, which is sampled from the test set of ACE 2005. From the EAE results, we can see that the vanilla DMCNN correctly classifies most of the event argument candidates. But because "sodomy" is a rare word, it misclassified "sodomy" into "N/A" (not an event argument). With the help of our NGS method's ability to model the joint distribution among event arguments, NGS (CNN) can infer that "sodomy" is a crime from the event argument correlations as it has known there are some crime-related arguments (adjudicator and plaintiff) in the sentence.

On the other side, we show the comparisons between the basic model DMCNN and NGS (CNN) on data with different numbers of event arguments in Table 6. With the increase of event argument number, our improvements significantly rise, which demonstrates our improvements come from modeling the correlations among event arguments. Note that the F1 scores are higher than the overall F1 scores, which is due to we filter out the negative instances without event arguments.

5 Conclusion and Future Work

In this paper, we propose a novel Neural Gibbs Sampling (NGS) method to adequately model the correlation between event arguments and argument roles, which combines the advantages of the Gibbs sampling method to model the joint distribution among random variables and the neural network models to automatically learn the effective representations. Considering the shortcoming of high complexity of Gibbs sampling algorithm, we further apply simulated annealing to accelerate the whole estimation process, which lead our method to being both effective and efficient.

The experimental results on two widely-used real-world datasets show that NGS can achieve comparable results to existing state-of-the-art EAE methods. The empirical analyses and ablation studies further verify the effectiveness and efficiency of our method. In the future: (1) We will try to extend NGS to other tasks and scenarios to evaluate its general effectiveness of modeling the latent correlations. (2) We will also explore more effective and simple methods to consider the correlations.

Acknowledgement

We thank Hedong (Ben) Hou for his help in the mathematical proof. This work is supported by the Key-Area Research and Development Program of Guangdong Province (2019B010153002), NSFC Key Projects (U1736204, 61533018), a grant from Institute for Guo Qiang, Tsinghua University (2019GQB0003) and THUNUS NExT Co-Lab. This work is also supported by the Pattern Recognition Center, WeChat AI, Tencent Inc. Xiaozhi Wang is supported by Tsinghua University Initiative Scientific Research Program.

References

David Ahn. 2006. The stages of event extraction. In *ARTE*.

Jun Araki and Teruko Mitamura. 2015. Joint event trigger identification and event coreference resolution with structured perceptron. In *EMNLP*.

P Basile, A Caputo, G Semeraro, and L Siciliani. 2014. Extending an information retrieval system through time event extraction. In *DART*.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *ACL*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL-IJCNLP*.

Pengxiang Cheng and Katrin Erk. 2018. Implicit argument prediction with event knowledge. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. Exploiting document level information to improve event detection via recurrent neural networks. In *IJCNLP*.

Greg Dubbin, Archna Bhatia, Bonnie Dorr, Adam Dalton, Kristy Hollingshead, Suriya Kandaswamy, Ian Perera, and Jena D Hwang. 2016. Improving discern with deep learning. In *TAC*.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M Strassel. 2015. Overview of linguistic resources for the tac kbp 2016 evaluations: Methodologies and results. In *TAC*.

James Ferguson, Colin Lockard, Natalie Hawkins, Stephen Soderland, Hannaneh Hajishirzi, and Daniel S Weld. 2016. University of washington tac-kbp 2016 system description. In *TAC*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*.

Stuart Geman and Donald Geman. 1987. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in computer vision*.

Reza Ghaeini, Xiaoli Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *ACL*.

Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *ACL-IJCNLP*.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *ACL-HLT*.

Yu Hong, Wenxuan Zhou, Guodong Zhou, Qiaoming Zhu, et al. 2018. Self-regulation: Employing a generative adversarial network to improve event detection. In *ACL*.

Andrew Hsi, Jaime G Carbonell, and Yiming Yang. 2016. Cmu cs event tac-kbp2016 event argument extraction system. In *TAC*.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *ACL*.

Ruihong Huang and Ellen Riloff. 2012a. Bootstrapped training of event extraction classifiers. In *Proceedings of EACL*.

Ruihong Huang and Ellen Riloff. 2012b. Modeling textual cohesion for event extraction. In *AAAI*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *ArXiv*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*.

Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. 1983. Optimization by simulated annealing. *Science*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*.

Shasha Liao and Ralph Grishman. 2010a. Filtered ranking for bootstrapping in event extraction. In *COLING*.

Shasha Liao and Ralph Grishman. 2010b. Using document level cross-event inference to improve event extraction. In *ACL*.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018. Nugget proposal networks for chinese event detection. In *ACL*.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. In *AAAI*.

Shaobo Liu, Rui Cheng, Xiaoming Yu, and Xueqi Cheng. 2018b. Exploiting contextual information via dynamic memory network for event detection. In *EMNLP*.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016a. Leveraging framenet to improve automatic event detection. In *ACL*.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *ACL*.

Shulin Liu, Kang Liu, Shizhu He, and Jun Zhao. 2016b. A probabilistic soft logic based approach to exploiting latent and global information in event classification. In *AAAI*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*.

David McClosky, Mihai Surdeanu, and Christopher D Manning. 2011. Event extraction as dependency parsing. In *ACL*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*.

Thien Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL-HLT*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *ACL-IJCNLP*.

E. Nummerlin. 1984. *General Irreducible Markov Chains and Non-negative Operators*.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *EMNLP*.

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML-PKDD*.

Lei Sha, Jing Liu, Chin-Yew Lin, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. RBPB: Regularization-based pattern balancing method for event extraction. In *ACL*.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *AAAI*.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of EMNLP-IJCNLP*, pages 313–325.

Liang Sun, Jason Mielens, and Jason Baldridge. 2014. Parsing low-resource languages using Gibbs sampling for PCFGs with latent annotations. In *EMNLP*.

L. Tierney. 1991. Ace 2005 multilingual training corpus. *Tech. Rept., School of Statist., Univ. of Minnesota*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of EMNLP-IJCNLP*, pages 5784–5789.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019a. Adversarial training forweakly supervised event detection. In *NAACL-HLT*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019b. HMEAE: Hierarchical modular event argument extraction. In *EMNLP-IJCNLP*.

Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. In *NAACL-HLT*.

Hui Yang, Tat-Seng Chua, Shuguang Wang, and Chun-Keat Koh. 2003. Structured use of external knowledge for event-based open domain question answering. In *SIGIR*.

Ying Zeng, Yansong Feng, Rong Ma, and Zheng Wang. 2018. Scale up event extraction learning via automatic training data generation. In *AAAI*.

Tongtao Zhang, Spencer Whitehead, Hanwang Zhang, Hongzhi Li, Joseph Ellis, Lifu Huang, Wei Liu, Heng Ji, and Shih-Fu Chang. 2017. Improving event extraction via multimodal integration. In *MM*.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *ACL*.

A Proof of the Convergence of Gibbs Sampling

In this section, we will prove the convergence of Gibbs sampling, by which we implement sampling from the implicit joint distribution in this paper.

Suppose that $X = (X_0, \cdots, X_n, \cdots)$, $X_i \in E \subseteq \mathbf{R}^n$ is a Markov chain (abbr. MC). For a ν-measurable set A, the transition kernel of A, $K : E \times E \to \mathbf{R}^n$ is defined via the following equation,

$$K(X_i, A) = \mathbf{P}\left\{X_{i+1} \in A | X_0, \cdots, X_i\right\} \quad (1)$$

Assume that X satisfies that for any σ-finite Borel measure ν on $\mathbf{R}^n$, for any ν-measurable set A, we have that,

$$\mathbf{P}\left(X_i \in A | X_{i-1} = x\right) = \int_A K(x, y)\mathrm{d}\nu(y) + \chi_A(x)r(x) \quad (2)$$

where

$$r(x) := 1 - \int_E K(x, y)\mathrm{d}\nu(y)$$

A fundamental property of K is sub-stochastic. Assume that K is non-degenerate, hence $r(x) < 1$ for all $x \in E$. Then, following the convention, we can define the iterative form as,

$$K^{(t)}(x, y) = \int_{\mathbf{R}^n} K^{(t-1)}(x, z)K(z, y)\mathrm{d}\nu(z) + K^{(t-1)}(x, y)r(y) + [1 - r(x)]^{t-1}K(x, y) \quad (3)$$

Define the invariant distribution as $\pi(X)$ for this MC and $D = \{x \in E; \pi(x) > 0\}$. We know that $\pi(X)$ must satisfy that, for any ν-measurable set A,

$$\pi(A) = \int P\left(X_1 \in A | X_0 = x\right)\pi(x)\mathrm{d}\nu(x) \quad (4)$$

For ν-measurable A, K is called ***π-irreducible*** when for all $x \in D, \pi(A) > 0$, and is called ***aperiodic*** when there exists no partition $E = (E_1, \cdots, E_{k-1})$ such that $\mathbf{P}(X_{i+1} \in A_{j+1} | X_i \in A_j) = 1$ for all $j = 1, \cdots, k-1 \pmod k$. Due to the work of Nummerlin (1984) and Tierney (1991), we have the following theorem: If K is π-irreducible and aperiodic then, for all $x \in D$.

1. $\left|K_x^{(t)} - \pi\right| \to 0$ as $t \to \infty$;
2. for real-valued, π-integrable function f,

$$t^{-1}\left\{f(X_1) + \cdots + f(X_t)\right\} \to \int_E f(x)\pi(x)d\nu(x) \text{ a.s. as } t \to \infty$$

where following the conventional transformation between multi-variable functions and parameter families, $K_x^{(t)}$ is defined as $K_x^{(t)}(y) := K^{(t)}(x, y)$. Indeed, with respect to ν, it is the density of X_t provided that $X_0 = x$, excluding the realizations $X_j = x, j = 1, \cdots, t$.

Let $\mathbf{P}(\mathbf{X}) = \mathbf{P}(X_1, \cdots, X_n)$ denote the target density in our case. What we shall prove is that this $\mathbf{P}(\mathbf{X})$ is the invariant distribution of the MC constructed by Gibbs sampling. Provided with the theorem above, the remaining key issue is to prove that the transition kernel K satisfies π-irreducibility and aperiodicity.

Equipped with the product measure, for the blocking $x = (x_1, \cdots, x_n)$, it is required that the conditionals of Gibbs sampler construction,

$$\pi(x_i|x_{-i}) = \frac{\pi(x)}{\int \pi(x) d\nu_i(x_i)}$$

are well-defined over the appropriate regions, where $\mathbf{X}_{-i}$ shares the same definition as Sec.(2). With $D = \{x \in E; \pi(x) > 0\}$, we seek to construct the kernel as $K : D \times D \to \mathbf{R}^n$ via

$$K(x, y) = \begin{cases} \prod_{i=1}^{n} (\pi(y_i|x_{j,j>i}, y_{j,j<i})) & \text{if } \Upsilon \\ 0 & \text{otherwise} \end{cases}$$

where Υ denotes the condition that

$$\pi(y_1, \cdots, y_i, x_{i+1}, \cdots, x_n) d\nu_i(y_i) > 0$$

It is then straightforward to check that, when $K(x, y)$ is well-defined, π is an invariant distribution of the chain attained by K.

Observe that since we have a discrete distribution, it is trivial that all the subjects here are well-defined. Also the aperiodicity of K is ensured by the fact that $K(x, x) > 0$ for all $x \in D$.