Minimize Exposure Bias of Seq2Seq Models in Joint Entity and Relation Extraction

Ranran Haoran Zhang[*1]**, Qianying Liu**[*2]**, Aysa Xuemo Fan**[1]**, Heng Ji**[1]**, Daojian Zeng**[4]**,**
Fei Cheng[2]**, Daisuke Kawahara**[3] **and Sadao Kurohashi**[2]

[1] University of Illinois at Urbana-Champaign
[2] Graduate School of Informatics, Kyoto University
[3] School of Fundamental Science and Engineering, Waseda University
[4] Hunan Normal University

{haoranz6,xuemof2,hengji}@illinois.edu; ying@nlp.ist.i.kyoto-u.ac.jp
{feicheng,kuro}@i.kyoto-u.ac.jp; dkw@waseda.jp; zengdj916@163.com

Abstract

Joint entity and relation extraction aims to extract relation triplets from plain text directly. Prior work leverages Sequence-to-Sequence (Seq2Seq) models for triplet sequence generation. However, Seq2Seq enforces an unnecessary order on the unordered triplets and involves a large decoding length associated with error accumulation. These methods introduce exposure bias, which may cause the models overfit to the frequent label combination, thus limiting the generalization ability. We propose a novel Sequence-to-Unordered-Multi-Tree (Seq2UMTree) model to minimize the effects of exposure bias by limiting the decoding length to three within a triplet and removing the order among triplets. We evaluate our model on two datasets, DuIE and NYT, and systematically study how exposure bias alters the performance of Seq2Seq models. Experiments show that the state-of-the-art Seq2Seq model overfits to both datasets while Seq2UMTree shows significantly better generalization. Our code is available at https://github.com/WindChimeRan/OpenJERE.

1 Introduction

Relation extraction aims to extract entity-relation triplets (h, r, t) from plain text. For example, in the triplet *(Obama, graduate_from, Columbia University)*, *Obama* and *Columbia University* are the head and tail entities appearing in the text, and *graduate_from* is the relation between these two entities. For supervised relation extraction, early studies focus on pipeline methods, which use an entity extractor to extract entities, and then classify the relations of entity pairs. These methods ignore the intrinsic interactions between these two subtasks and propagate classification errors through the tasks. Jointly entity and relation extraction (JERE) considers the subtask interaction (Roth and Yih, 2004;

Ji and Grishman, 2005; Ji et al., 2005; Yu and Lam, 2010; Riedel et al., 2010; Sil and Yates, 2013; Li et al., 2014; Li and Ji, 2014; Durrett and Klein, 2014; Miwa and Sasaki, 2014; Lu and Roth, 2015; Yang and Mitchell, 2016; Kirschnick et al., 2016; Miwa and Bansal, 2016; Gupta et al., 2016; Katiyar and Cardie, 2017) , but they mainly exploit feature-based system or multi-task neural network, which can not capture inter-triplet dependency.

NovelTagging (Zheng et al., 2017) integrates these two subtasks into one sequence labeling process, which assigns a single entity-relation tag to each token; when a token belongs to multiple relations, the prediction results will be incomplete. Instead of sequence labeling, Sequence-to-Sequence (Seq2Seq) models (Cho et al., 2014) are able to extract an entity multiple times, thus multiple relations can be assigned to one entity, which solves the problem naturally (Zeng et al., 2018, 2019a,b; Nayak and Ng, 2019). Specifically, all existing Seq2Seq models pre-define a sequential order for the target triplets, e.g. triplet alphabetical order, and then decode the triplet sequence according to the order autoregressively, which means the current triplet prediction relies on the previous output. For exmaple, in Figure 1, the triplet list is flattened to *[Obama]-[graduate_from]-[Columbia University]-[Obama]-[graduate_from]-[Harvard Law School]...*

However, the autoregressive decoding of the Seq2Seq models introduces exposure bias problem which may severely reduce the performance. Exposure bias refers to the discrepancy between training and testing phases of the decoding process (Ranzato et al., 2015). In the training phase, the current triplet prediction relies on the gold-standard labels of the previous triplets, while in the testing phase, the current triplet prediction relies on the model prediction of the previous triplets, which can be different from the gold-standard labels. As a result,

---
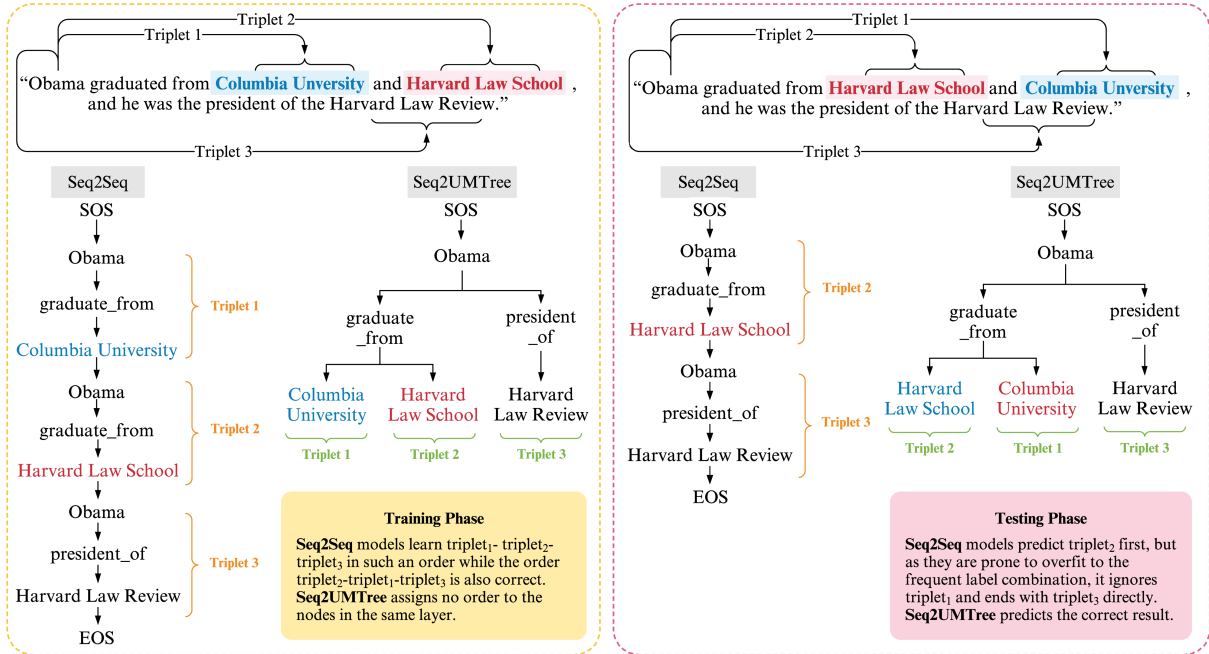* This denotes equal contribution.

236

Figure 1: The training and testing of Seq2Seq and Seq2UMTree for different triplet orders.

in the test phase, a skewed prediction will further deviate the predictions of the follow-up triplets; if the decoding length is large, the discrepancy from the gold-standard labels would be further accumulated. Such accumulated discrepancy may decrease the performance especially in predicting longer sequences, i.e., multi-triplet prediction.

Furthermore, because Seq2Seq model sequentially predicts the triplets, it enforces an unnecessary order on the unordered labels, while other triplet orders are also correct. Thus, the assigned order makes the model prone to memorize and overfit to the frequent label combinations in the training set and poorly generalize to the unseen orders. The overfitting is also the side effect of exposure bias (Tsai and Lee, 2019), which may result in missing triplets in Seq2Seq prediction. For example, in Figure 1, during the training phase, the Seq2Seq model learns $triplet_1$-$triplet_2$-$triplet_3$ in such an order while the order $triplet_2$-$triplet_1$-$triplet_3$ is also correct. In the testing phase, the Seq2Seq model predicts $triplet_2$ first based on the assigned order, but because $triplet_2$-$triplet_3$ is a frequent order for the model, it ignores $triplet_1$ and ends with $triplet_3$ directly (i.e.,$triplet_2$-$triplet_3$). When an order is enforced on the model, the model proceeds with more learning constrains.

To mitigate the exposure bias problem while keeping the simplicity of Seq2Seq, we recast the one-dimension triplet sequence to two-dimension Unordered-Multi-Tree (UMTree) and propose a novel model **Seq2UMTree**. The Seq2UMTree model is based on an Encoder-Decoder framework, which is composed of a conventional encoder and a UMTree decoder. The UMTree decoder models entities and relations jointly and structurally, using a copy mechanism with unordered multi-label classification as the output layer. This multi-label classification model ensures the nodes in the same layer are unordered and discards the predefined triplet order so that the prediction deviation will not aggregate and affect other triplets. Different from the standard Seq2Tree (Dong and Lapata, 2016; Liu et al., 2019), the decoding length is limited to three (one triplet), which is the shortest feasible length for JERE task. In this way, the exposure bias is minimized under the triplet-level F1 metrics.

In conclusion, our contributions are listed as follows:

- We point out the redundancy of the predefined triplet order of the Seq2Seq model, and propose a novel Seq2UMTree model to minimize exposure bias by recasting the ordered triplet sequence to an Unordered-Multi-Tree format.

- We systematically analyze how exposure bias diminishes the reliability of the performance scores of the standard Seq2Seq models.
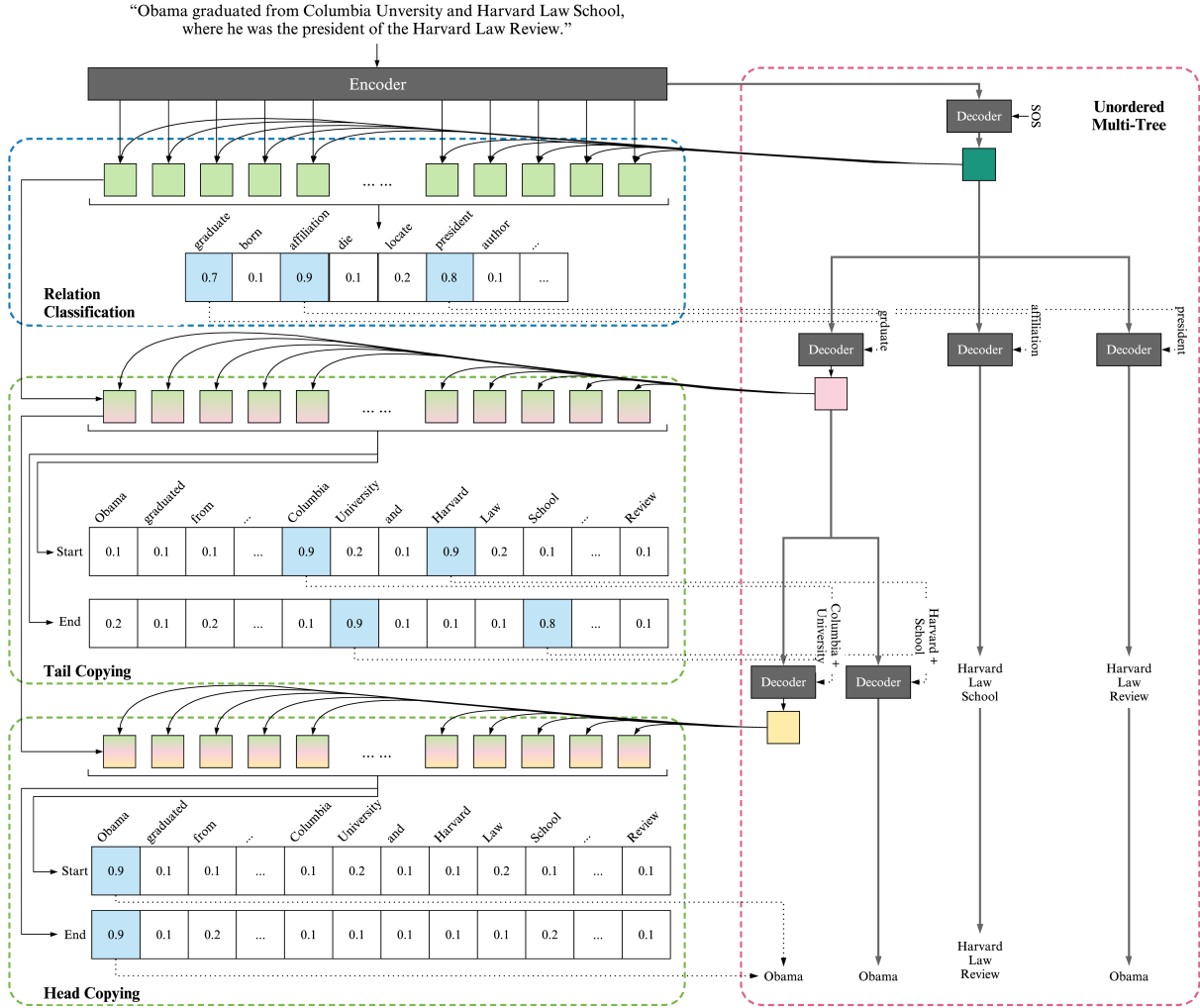
237

Figure 2: The model overview. The decoding order within a triplet is r, t, h. The relation is predicted from a predefined relation dictionary and the entities are copied from the sentence.

2 Methodology

The Seq2UMTree model consists of a conventional Seq2Seq encoder and a UMTree decoder. The UMTree decoder is different from the standard decoder in a way that it generates unordered multi-label outputs and uses the UMTree decoding strategy. The overview of the model is shown in Figure 2. We illustrate the model details in the following subsections.

2.1 Model

Formally, the input sentence $\boldsymbol{x} = [x_0, x_1, \ldots, x_n]$ is first transformed to a sequence of context aware representations by word embedding and Bidirectional Recurrent Neural Network (Bi-RNN) (Schuster and Paliwal, 1997) with Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997)

as the encoder:

$$[\boldsymbol{s}_0^E, \boldsymbol{s}_1^E, \ldots, \boldsymbol{s}_n^E] = \text{Encoder}([x_0, x_1, \ldots, x_n]) \quad (1)$$

Then we pass the output $\boldsymbol{s}$ sequence to Conv_{en}:

$$\boldsymbol{o}_0 = \text{Conv}_{en}([\boldsymbol{s}_0^E, \boldsymbol{s}_1^E, \ldots, \boldsymbol{s}_n^E]) \quad (2)$$

where Conv_{en} is the encoder convolutional layer. Conv_{en} maps $\boldsymbol{s}^E$ to $\boldsymbol{o}_0$, which is also a sequence and has the identical dimension as the $\boldsymbol{s}$ sequence. The output is denoted as $\boldsymbol{o}_0 \in \mathbb{R}^{n \times h}$, where h is the hidden size, n is the length of the input sentence. $\boldsymbol{o}_0$ is the auxiliary representation of the sentence, which is used for decoding with scratchpad attention mechanism (Benmalek et al., 2019): $\boldsymbol{o}_{n-1}$ is used to calculate attention score, and $\boldsymbol{o}_{n-1}$ will be updated to $\boldsymbol{o}_n$ at every decoding step.

During decoding, we use different input embeddings and output layers for relation and entity ex-
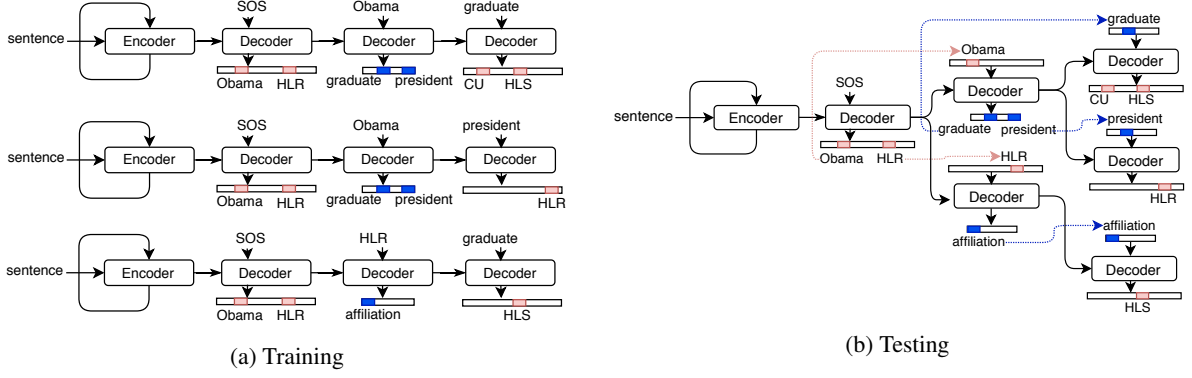
238

(a) Training

(b) Testing

Figure 3: Seq2UMTree is trained in a teacher-forcing way by aligning the tree to the sequences. In the test phase, the model decodes the whole tree autoregressively. In the figure, HLR, HLS, CU are the abbreviations of Harvard Law Review, Harvard Law School and Columbia University. The example uses *h-r-t* as the order within a triplet.

traction, and they share the same decoder parameters. For the input embedding $\boldsymbol{w}_t$, we use: (a) "start-of-the-sentence" embedding: $\boldsymbol{w}_0^{sos} \in \mathbb{R}^h$, which is always the beginning of the decoding and is considered as depth 0, (b) relation embedding: $\boldsymbol{w}_t^r \in \mathbb{R}^h$, (c) entity embedding: $\boldsymbol{w}_t^e = \boldsymbol{o}_{t-1}^{e1} + \boldsymbol{o}_{t-1}^{e2} \in \mathbb{R}^h$, where $e1$ and $e2$ are the beginning position and the end position of the predicted entity respectively. $t \in \{1, 2, 3\}$, which is the decoding time step. The decoding order can be predefined arbitrarily, such as h-r-t or t-r-h.

Given the input embedding $\boldsymbol{w}_t$ and the output of the previous time step $\boldsymbol{s}_{t-1}^D$, a unary LSTM decoder is used to generate decoder hidden state:

$$\boldsymbol{s}_t^D = \text{Decoder}(\boldsymbol{w}_t, \boldsymbol{s}_{t-1}^D) \qquad (3)$$

where $\boldsymbol{s}_t^D$ is the decoder hidden states; $\boldsymbol{s}_0^D$ is initialized by $\boldsymbol{s}_n^E$.

Attention mechanism (Luong et al., 2015) is used to generate context-aware embedding:

$$\boldsymbol{a}_t = \text{Attention}(\boldsymbol{o}_{t-1}, \boldsymbol{s}_t^D) \qquad (4)$$

where $a \in \mathbb{R}^h$. Then the context-aware representation $\boldsymbol{a}_t$ is concatenated with the original $\boldsymbol{o}_{t-1}$, followed by a convolution layer:

$$\boldsymbol{o}_t = \text{Conv}_{de}([\boldsymbol{a}_t; \boldsymbol{o}_{t-1}^{0:n}]) \qquad (5)$$

where Conv_{de} maps dimension $2h$ to h and $\boldsymbol{a}_t$ is replicated n times before concatenation.

The output layer of the relation prediction is a linear transformation followed by a max-pooling over sequence:

$$\boldsymbol{prob}_r = \sigma(\text{Max}(\boldsymbol{o}_t \boldsymbol{W}_r + \boldsymbol{b}_r)) \qquad (6)$$

where σ is the sigmoid function for multi-relation classification, $\boldsymbol{W}_r \in \mathbb{R}^{h \times r}$, $\boldsymbol{b}_r \in \mathbb{R}^r$ and $\boldsymbol{prob}_r \in \mathbb{R}^r$ is the predicted probability vector of the relations.

The output layers of the entity prediction are two binary classification layers over the whole sequence, predicting the positions of the beginning and the end of the entities respectively:

$$\begin{aligned} \boldsymbol{prob}_{e_b} &= \sigma(\boldsymbol{W}_{e_b}^T \boldsymbol{o}_t + b_{e_b}) \\ \boldsymbol{prob}_{e_e} &= \sigma(\boldsymbol{W}_{e_e}^T \boldsymbol{o}_t + b_{e_e}) \end{aligned} \qquad (7)$$

where $\boldsymbol{W}_e \in \mathbb{R}^{h \times 1}$, b_e is a scalar and $\boldsymbol{prob}_e \in \mathbb{R}^{n \times 1}$ is the predicted probability vector of the entities, e_b and e_e refer to the beginning and the ending of the entity. Different from Nayak and Ng (2019), the sigmoid function σ enables the model to predict multiple entities at a time.

2.2 Training and Testing

In the training phase, for each sentence, we reorganize the training data that each pair of depth 1 and 2 (e.g. h-r) in UMTree would form one training example, so that this strategy traverses the whole tree. The training process of each node corresponds to one time step in Seq2Seq models. We then train the model in teacher forcing (Williams and Zipser, 1989) manner: the input of each decoding time step is given by the gold-standard labels. Take the order h-r-t as an example, in Fig. 3a, the total loss is the sum of the losses of the following three decoding

239

	NYT				DuIE			
	test#	Prec	Rec	F1	test#	Prec	Rec	F1
CopyMTL	.978	.685	.648	.666	.962	.496	.394	439
WDec	.988	**.843**	**.764**	**.802**	.919	.641	.542	.587
MHS	.995	.798	.739	.768	.984	**.772**	.623	.690
Seq2UMTree	1.00	.791	.751	.771	1.00	.756	**.730**	**.743**

Table 1: Main Results on NYT and DuIE. #test is the valid sentence percentage of the test set to the models.

	NYT		DuIE	
	#sentence	#triplet	#sentence	#triplet
train	56,195	90,967	155,931	314,996
dev	5,000	8,153	17,178	34,270
test	5,000	8,214	21,639	43,749

Table 2: Data statistics of NYT and DuIE datasets. NYT contains 24 relations and DuIE contains 49 relations.

steps:

$$L = -\log \Pr(h_b = h_b^*|\boldsymbol{x};\theta)$$
$$- \log \Pr(h_e = h_e^*|\boldsymbol{x};\theta)$$
$$- \log \Pr(r = r^*|h_b^*, h_e^*, \boldsymbol{x};\theta) \qquad (8)$$
$$- \log \Pr(t_b = t_b^*|r^*, h_b^*, h_e^*, \boldsymbol{x};\theta)$$
$$- \log \Pr(t_e = t_e^*|r^*, h_b^*, h_e^*, \boldsymbol{x};\theta)$$

where h^*, r^*, t^* are the ground truth of the triplets, θ is all of the trainable parameters in the model. In the testing phase, the UMTree uses auto-regressive decoding strategy. The decoder predicts the nodes layer by layer, where the prediction results of the previous layer are used as the input of the next time step separately, as shown in Fig. 3b.

3 Experiments

3.1 Settings

Dataset

We evaluate our model on two datasets, NYT and DuIE[1]. NYT (Riedel et al., 2010) is a English news dataset that is generated by distant supervision without manual annotation, which is widely used in JERE studies (Zheng et al., 2017; Zeng et al., 2018; Takanobu et al., 2018; Dai et al., 2019; Fu et al., 2019; Nayak and Ng, 2019; Zeng et al., 2019a,b; Chen et al., 2019; Wei et al., 2019). We use the same data split as CopyRE (Zeng et al., 2018). DuIE (Li et al., 2019) is a large-scale Chinese JERE dataset where sentences are from Baidu

News Feeds and Baidu Baike. The whole dataset is annotated by distant supervision and then checked manually. We take 10% of the training set randomly as a validation set and the original development set as the test set because the original test set is not released. In prerprocessing, for both datasets, we filter out the sentences that contain no triplet. The data statistics of these two datasets are shown in Table 2.

Baselines

We compare the proposed model, Seq2UMTree, with strong baselines under the same hyperparameters, as follows: 1) CopyMTL (Zeng et al., 2019a) is a Seq2Seq model with copy mechanism, and the entities are found by multi-task learning. 2) WDec (Nayak and Ng, 2019) is a standard Seq2Seq model with dynamic masking, and decode the entity token by token. 3) MHS (Bekoulis et al., 2018) is a non-Seq2Seq baseline, which enumerates all possible token pairs. 4) Seq2UMTree is the proposed method, which generates triplets in a concise tree structure.

Hyperparameters

For the sake of fair comparison, we reproduce all the baselines ourselves with the same hyperparameter settings. We use 200-dimension word embedding for English and character embedding for Chinese. Both are initialized from Gaussian distribution $\mathcal{N}(0,1)$, and 200-dimension Bi-LSTM encoder is used for both to mitigate the heterogeneity of these two languages. These models are trained for 50 epochs by Adam optimizer (Kingma and Ba, 2014), and the models with the highest validation F1 scores are used for testing. The training of all compared models can be finished in 24 hours in a single NVIDIA V100 16GB GPU. The decoding order of Seq2UMTree in both datasets is r-t-h. We will discuss the effect of the order in subsection 4.2.

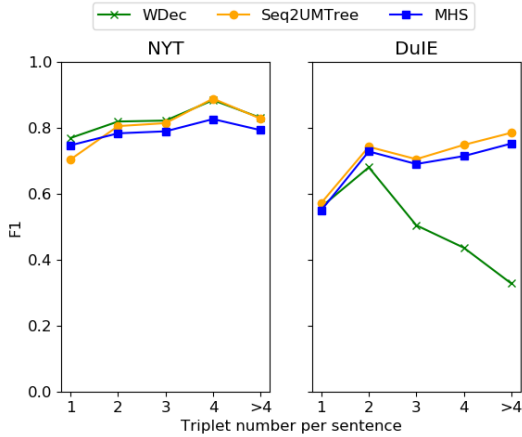[1] https://ai.baidu.com/broad/introduction?dataset=dureader

Figure 4: The F1 scores of the models on test subsets NYT and DuIE with different numbers of triplets. The subsets contain sentences with number of triplets 1, 2, 3, 4 and >4 have 3080, 1127, 298, 315 and 470 in NYT and 9853, 7034, 2366, 1153 and 1233 in DuIE.
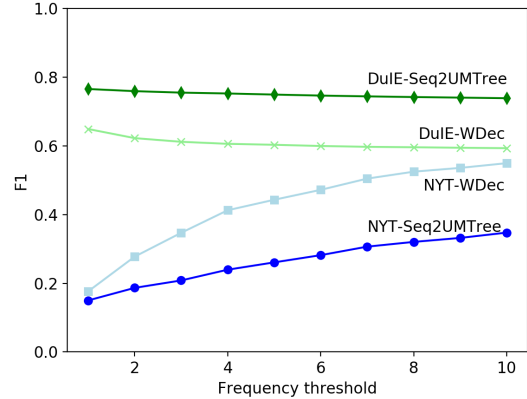

Figure 5: The F1 scores of the models with triplet frequency less than threshold. Triplet frequency represents how often the test triplets appeared in the training set.

3.2 Main Results

The experiment results are shown in Table 1. Because of the limitation of GPU memory, the Seq2Seq models and MHS cannot process all the testing data. The valid sentence percentage of the test set is shown in the #test column. WDec sets the maximal decoding length to 50 and CopyMTL can only decode 5 triplets at most, resulting in their incomplete coverage on DuIE testset, in which 8.1% and 3.8% of the test sentences are deleted in the preprocessing stage. Moreover, because the entities in DuIE usually have more tokens than NYT does, the maximal decoding length of WDec filters out more examples in DuIE (8.1%) than in NYT (1.2%). MHS extracts triplets by exhaustively enumerating all token pairs, resulting in a $\mathcal{O}(l^2 r)$ GPU memory consumption of encoding sentences, where l is the sentence length and r is the number of relations. In our reproduction, we delete sentences longer than 100 tokens in NYT and 150 in DuIE, which covers 0.5% of the NYT test set and 1.6% of the DuIE test set. Among all the models, only Seq2UMTree can be applied for all sentences in both datasets[2] and the space complexity is $\mathcal{O}(2l + r)$.

From the Table 1 we can see that Seq2UMTree outperforms the previous best Seq2Seq model WDec by 15.6% F1 score in DuIE, but it underperforms WDec in NYT by 3.1%. The inconsistency of the performances on two datasets motivates us

to conduct deeper investigation in the next section.

4 Investigation on Data & Model Bias

4.1 Exposure Bias and Generalization

While Seq2Seq assigns an order to the triplets, Seq2UMTree generates triplets in an unordered way, regardless of the triplet number. To verify the effectiveness of Seq2UMTree on multiple triplets, we split NYT and DuIE test sets into five subsets in which each sentence only contains a specific number of triplets (1, 2, 3, 4, >4). The performance of the models in the subsets is shown in Figure. 4. In DuIE, when the triplet number increases, the F1 scores of WDec decrease drastically from 70% to 40% for triplet numbers greater than 2. MHS and Seq2UMTree perform better as the triplet number increases. By contrast, in NYT, all models perform similarly with different numbers of triplets. To better address the reasons behind the performance differences, we conduct qualitative analysis of the data, finding that in NYT, 90% triplets in the test set reoccurred in the training set, while in DuIE, the percentage is only 30%. Based on this observation, we hypothesize that the Seq2Seq models gain high score in NYT because of exposure bias: as the triplets in the test set are highly overlapped with those in the training set, the models achieve high scores by memorizing the frequently reoccurred training set triplets, which causes the overfitting that makes the models generalize poorly to the unseen triplets.

To investigate the effects of data bias from reoccurred frequency, we split the test set into 10 sub-

---

[2]The performance scores are calculated in their processed test sets.

		Prec	Rec	F1
Test-A	Seq2UMTree	.891	.882	.886
	WDec	.956	.862	.906
Test-B	Seq2UMTree	.695	.579	.631
	WDec	.616	.562	.588

Table 3: AB-Test on NYT. We split NYT test set to two subsets. The triplets in Test-A set (2,625 sentences) 100% have occurred in the filtered training subset (33,963 sentences) while the triplets in Test-B set (2,317 sentences) have never occurred in the filtered training set.

sets according to the reoccurred frequency (1-10) of triplets in the training set. The results are shown in Figure. 5. In NYT, the F1 scores of both WDec and Seq2UMTree increases as the reoccurred frequency increases. In DuIE, the performance curve is almost flat despite of the reoccurred frequency. This implies that the performance is highly related to the reoccurred frequency in NYT (90% reoccurred) but is minimally related to that in DuIE (30% reoccurred).

To further testify the effects of exposure bias on seen and unseen data, we conduct an AB test on the NYT dataset. We take a new training set from the NYT training set, and then take two new test sets, Test-A and Test-B, from the NYT test set: Test-A's triplets is 100% overlapped with these in the new training set but the triplets in Test-B have never appeared in the new training set. The new training set consists of 60% of the original. Test-A and Test-B contain 53% and 47% of the sentences from the original, respectively. The results are reported in Table. 3.

Though Seq2UMTree underperforms WDec in 100%-overlapped set, it outperforms WDec in unseen set. The performance drop (from seen to unseen) for Seq2UMTree is smaller than WDec's, which implies that Seq2UMTree is more robust and more reliable. This verifies our hypothesis that the Seq2Seq models suffer more from exposure bias, which results in more overfitting, while Seq2UMtree with minimized exposure bias is more generalized to the unseen triplets.

As the NYT dataset intrinsically has high portion of overlapped triplets in its training and test sets, and has already been overfitted by existing models, we suggest that NYT is not unbiased enough to be used as a baseline dataset, and the F1 scores of the models on NYT are not reliable.

	Order	Prec	Rec	F1
NYT	t, r, h	.788	.694	.738
	r, t, h	.791	**.751**	**.771**
	t, h, r	.765	.495	.601
	h, t, r	.756	.548	.635
	r, h, t	.789	.737	.762
	h, r, t	**.796**	.685	.737
DuIE	t, r, h	.766	.663	.711
	r, t, h	.756	**.730**	**.743**
	t, h, r	**.802**	.330	.467
	h, t, r	.794	.120	.208
	r, h, t	.760	.712	.735
	h, r, t	.731	.728	.729

Table 4: Different orders of Seq2UMTree.

4.2 Orders within Triplets

In Seq2UMTree, the relation, head entity and tail entity are still decoded in a predefined order (e.g., h-r-t or r-t-h). We enumerate all six possible decoding orders in each dataset and compare the performances. The results are shown in Table. 4. The performances varies by order within triplets, while the recalls for orders t-h-r and h-t-r drop drastically in both datasets, respectively.

We then hypothesize that the order within the triplets matters in some way. Thinking of this, we decide to look into the training phase time step by time step, and find that these 2 orders cannot even fit training set well: the recall for h-t-r is only 13% on the training set (12% on the test set). Moreover, most of the the predictions are missing on the first time step (h). This implies that the position of r provides information important to the predictions and proved our hypothesis. By thorough error analysis, we realize that for the order h-t-r (t-h-r follows the same logic), the model has to predict all t with regard to h in the second time step, without constraints from the r, and this makes every possible entity to be a prediction candidate. However, the model is unable to eliminate no-relation entity pairs at the third time step, thus the model is prone to feed entity pairs to the classification layer with an low odds (low recall) but high confidence (high precision).

In contrast, for the order h-r-t, given the predicted h, the corresponding r can be easily identified according to the context. Subsequently, the predicted h-r pair gives strong hint to the last time step prediction, hence the model will not collapse from the no-relation. This also applies to any other

order with r in the first two time steps.

5 Related Work

Previous work uses PIPELINE to extract triplets from text (Nadeau and Sekine, 2007; Chan and Roth, 2011). They first recognize all entities in the input sentence then classify relations for each entity pair exhaustively. Li and Ji (2014) point out that the classification errors may propagate across subtasks. Instead of treating these two subtasks separately, for joint entities and relations extraction (JERE), TABLE methods calculate the similarity score of all token pairs and relations by exhaustive enumeration and the extracted triplets are found by the position of the output in the table (Miwa and Bansal, 2016; Gupta et al., 2016). However, as a triplet may contain entities with different lengths, the table methods either suffer from exponential computational burden (Adel and Schütze, 2017) or roll back to pipeline methods (Sun et al., 2018; Bekoulis et al., 2018; Fu et al., 2019). Furthermore, such table enumeration dilutes the positive labels quadratically, thus aggravating the class-imbalanced problem. To model the task in a more concise way, Zheng et al. (2017) propose a NOVELTAGGING scheme, which represents relation and entity in one tag, so that the joint extraction can be solved by the well-studied sequence labeling approach. However, this tagging scheme cannot assign multiple tags to one token thus fail on overlapping triplets. The follow-on methods revise the tagging scheme to enable multi-pass sequence labeling (Takanobu et al., 2018; Dai et al., 2019) but they introduce a similar sparsity issue as does the table method.

Another promising method, SEQ2SEQ, is first proposed by Zeng et al. (2018). Seq2Seq does not only decode the triplet list straightforwardly but also circumvents the overlapping triplets problem. Although this paper introduces a problem that multi-token entities cannot be predicted, this problem has been solved by multiple follow-up papers (Zeng et al., 2019a; Nayak and Ng, 2019). However, there still remains a weakness in Seq2Seq models, i.e., the exposure bias, which has been overlooked.

Exposure bias originates from the discrepancy between training and testing: Seq2Seq models use data distribution for training and model distribution for testing (Ranzato et al., 2015). Existing work mainly focuses on how to mitigate the informa-

tion loss of $\arg\max$ sampling (Yang et al., 2018, 2019; Zhang et al., 2019). Nam et al. (2017) notice that different orders affect the performance of the Seq2Seq models in Multi-Class Classification (MCC), and conduct thoroughly experiments on frequency order and topology order. In JERE, Zeng et al. (2019b) study additional rule-based triplet prediction orders, including alphabetical, shuffle and fix-unsort, and then propose a reinforcement learning framework to generate triplets in adaptive orders dynamically. Tsai and Lee (2019) first point out the unnecessary order causes exposure bias altering the performance in MCC, and they find that Seq2Seq models are prone to overfit to the frequent label combination and show poor generalization on unseen target sequence.

Our method solves the exposure bias problem. As the exposure bias problem stems from the ordered left-to-right triplet decoding, we block the decoding of them from each other by removing the order of the triplet generation, thus the possible prediction error cannot propagate from triplet to triplet. Furthermore, because each triplet is generated by an independent decoding process, the decoding length has been extremely shortened, thus minimizes the effects of exposure bias. Our method differs from previous solution on exposure bias that we remove the order by structure decoding rather than random sampling (Tsai and Lee, 2019).

CASREL (Wei et al., 2020) is a recently proposed two-step tagging method, which first finds all the head entities in the sentence then labels a relation-tail table for each head entity, which can also be seen as a UMTree decoder with a decoding length two. However, they overlook the data bias problem in NYT, which causing model unreliability and possible model bias.

Note that our task is different from ONEIE (Lin et al., 2020), which models event extraction, entity span detection, entity type recognition and relation extraction in a Seq2Graph way. In contrast to ONEIE, JERE aims to extract only relation-entity triplets, which can be modeled by our UMTree structure naturally. The simplicity of the tree enables the model to conduct global extraction.

6 Conclusions

In this paper, we thoroughly analyze the effects of exposure bias of Seq2Seq models on joint entity and relation extraction. Exposure bias causes overfitting that hurts the reliability of the perfor-

mance scores. To solve the problem of exposure bias, we point out the order of the target triplets is redundant and formulate the target triplet sequence to Unordered-Multi-Tree. The Unordered-Multi-Tree structure minimizes the effect of exposure bias by limiting the decoding length to three within a triplet, and removing the order among triplets. We conduct in-depth experiments and reveal the relationship between exposure bias and data bias. The results show great generalization of our model.

References

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1723–1729, Copenhagen, Denmark. Association for Computational Linguistics.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Ryan Benmalek, Madian Khabsa, Suma Desu, Claire Cardie, and Michele Banko. 2019. Keeping notes: Conditional natural language generation with a scratchpad encoder. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4157–4167, Florence, Italy. Association for Computational Linguistics.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of ACL*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Jiayu Chen, Caixia Yuan, Xiaojie Wang, and Ziwei Bai. 2019. MrMep: Joint extraction of multiple relations and multiple entity pairs based on triplet attention. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 593–602, Hong Kong, China. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Haifeng Wang. 2019. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. *Proceedings of AAAI*, 33(01):6300–6308.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. In *Transactions of the Association for Computational Linguistics (TACL)*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Heng Ji and Ralph Grishman. 2005. Improving name tagging by reference resolution and relation detection. In *In Proceedings of ACL 05, Ann Arbor, USA*.

Heng Ji, David Westbrook, and Ralph Grishman. 2005. Using semantic relations to refine coreference decisions. In *In Proceedings of HLT/EMNLP 05, Vancouver, B.C., Canada*.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of ACL*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Johannes Kirschnick, Holmer Hemsen, and Volker Markl. 2016. JEDI: Joint entity and relation detection using type inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics System Demonstrations (ACL2016)*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Qi Li, Heng Ji, HONG Yu, and Sujian Li. 2014. Constructing information networks using one single model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*.

Shuangjie Li, Wei He, Yabing Shi, Wenbin Jiang, Haijin Liang, Ye Jiang, Yang Zhang, Yajuan Lyu, and Yong Zhu. 2019. Duie: A large-scale chinese dataset for information extraction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 791–800. Springer.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint end-to-end neural model for information extraction with global features. In *Proceedings of The 58th Annual Meeting of the Association for Computational Linguistics*.

Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. 2019. Tree-structured decoding for solving math word problems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2370–2379, Hong Kong, China. Association for Computational Linguistics.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Jinseok Nam, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. 2017. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5413–5423. Curran Associates, Inc.

Tapas Nayak and Hwee Tou Ng. 2019. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *arXiv preprint arXiv:1911.09886*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL2004)*.

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on Conference on Information & Knowledge Management (CIKM2013)*.

Changzhi Sun, Yuanbin Wu, Man Lan, Shiliang Sun, Wenting Wang, Kuang-Chih Lee, and Kewen Wu. 2018. Extracting entities and relations with joint minimum risk training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2265, Brussels, Belgium. Association for Computational Linguistics.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2018. A hierarchical framework for relation extraction with reinforcement learning. *arXiv preprint arXiv:1811.03925*.

Che-Ping Tsai and Hung-Yi Lee. 2019. Order-free learning alleviating exposure bias in multi-label classification. *arXiv preprint arXiv:1909.03434*.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2019. A novel hierarchical binary tagging framework for joint extraction of entities and relations. *arXiv preprint arXiv:1909.03227*.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging

framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488, Online. Association for Computational Linguistics.

R. J. Williams and D. Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT2016)*.

Pengcheng Yang, Fuli Luo, Shuming Ma, Junyang Lin, and Xu Sun. 2019. A deep reinforced sequence-to-set model for multi-label classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5252–5258, Florence, Italy. Association for Computational Linguistics.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xiaofeng Yu and Wai Lam. 2010. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Coling 2010: Posters*, pages 1399–1407, Beijing, China. Coling 2010 Organizing Committee.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2019a. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. *arXiv preprint arXiv:1911.10438*.

Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019b. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 367–377, Hong Kong, China. Association for Computational Linguistics.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.