Graph-to-Graph Annotation Conversion Based on Pretrained Models

Anonymous ACL submission

Abstract

Annotation conversion is an effective way to construct datasets under new annotation guidelines based on existing datasets with little human labour. Previous work has been limited in conversion between tree-structured datasets and mainly focused on feature-based models which are not easily applicable to new conversion. In this paper, we propose two pretrained model-based graph-to-graph annotation conversion approaches, namely Label Switching and Graph2Graph Linear Transfer, which are able to deal with conversion between graph-structured annotations and require no manually designed feature. We manually construct a graph-structured parallel annotated dataset and evaluate the proposed approaches on it as well as four existing parallel annotated datasets. Experimental results show that the proposed approaches outperform two strong baselines across all the datasets. Furthermore, the combination of the two models have a better effect.

1 Introduction

While tree-structured representations have dominated parsing for the last decade, graph-structured datasets are receiving growing interest in recent years (Oepen et al., 2019, 2020). Over the last few years, an increasing number of graph-structured datasets have become available. Some of them, such as DM corpora from the SemEval 2015 task 18 dataset (Oepen et al., 2015) and AMRBank (Banarescu et al., 2013), are manually annotated. While some others, such as the Enhanced English Universal Dependencies dataset (Schuster and Manning, 2016), are converted from existing datasets with manually designed rules. As illustrated in Figure 1, the Semantic Dependency Graph at the top is converted from the Universal Dependecy Tree at the bottom.

However, in the dataset construction process under a new annotation guideline, it would be extremely expensive to annotate the whole dataset manually. While rule-based conversion, although needs no human labour for annotation, requires expertise to design the rules, which could be difficult if the new guideline is vastly different from the old one. Therefore, it would be efficient and attractive to exploit existing dataset and learn a transformation that converts them into the new guideline. The converted dataset under the new guideline could be used in model training or further refined by human annotators to construct a high-quality dataset.

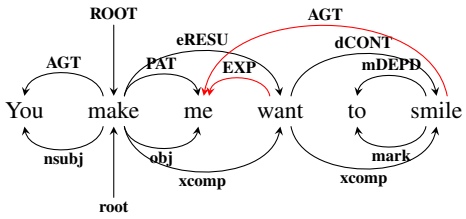

Figure 1: Example of annotation conversion from Universal Dependency Tree (bottom) to our Semantic Dependency Graph (top).

Such conversion has been studied in a line of research that exploits heterogeneous treebanks to boost parsing performance, where the approach is typically referred to as treebank conversion (Li et al., 2013; Jiang et al., 2018). In their case, two existing heterogeneous treebanks (tree-structured datasets) on different texts are available. The goal is to convert a source treebank into the target guideline and use the converted treebank as extra annotated data for the training of the target model. However, in our case, the goal is to construct a dataset under a new guideline. Therefore, only the source dataset and a small set of annotations under the target guideline are available. Besides, the approach should support conversion between graph-structured datasets rather than tree-structured ones.

Previous work for treebank conversion mainly focused on feature-based methods. Normally, they first construct parallel annotated data by manually annotating part of the target treebank under the

source guideline (Jiang et al., 2015, 2018) or training a parser on the source treebank and parsing the target treebank with it (Zhu et al., 2011; Li et al., 2013). Then they use the source annotations as extra guiding features to train an augmented target parser that parses the whole source treebank and generates the expected target annotations. Such methods are not easily applicable to new conversion since the annotation guidelines are normally vastly different from each other, and thus the features should be redesigned for every new guideline.

Pretrained models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), have achieved great success in a wide range of NLP tasks. Previous research has shown that such models are able to capture structural information implicitly (Jawahar et al., 2019; Lin et al., 2019), which could be helpful for the learning of transformation between graph-structured annotations.

Therefore, in this paper, we propose two pretrained model-based graph-to-graph annotation conversion approaches, namely Label Switching (LS) and Graph2Graph Linear Transfer (GGLT), which are able to deal with conversion between graph-structured datasets and require no manually designed feature. Specifically, in the Label Switching approach, we first automatically construct large scale pseudo target data by switching labels in source data to target labels based on the alignment information obtained from the parallel annotated data. After that, a pretrained model-based parser is first fine-tuned on the pseudo data and then further fine-tuned on the small set of gold target annotations. The parser is eventually used to parse the source dataset to generate the target annotation. The GGLT approach directly loads parameters from the parser trained on source annotation, then use a biaffine graph parser (Dozat and Manning, 2018) as a decoder to linearly transfer to the target annotation. We manually construct a graph-structured dataset under the refined semantic dependency graph (SDG) guideline (Che et al., 2016) on part of the text from the English Web Treebank (EWT) in the Universal Dependencies (UD) Treebanks (v2.5) (Zeman et al., 2019).[1] To verify the effectiveness of the proposed approaches, we further evaluate them on the conversion from UD-EWT to the Enhanced Universal Dependencies (UD-Enhanced) guideline (Schuster and Manning, 2016), conversion from the Universal Dependency Tree to the Semantic Dependency Graph and conversion between three types of annotations (i.e., DM, PAS and PSD) in the SemEval 2015 task 18 dataset (Oepen et al., 2015). Experimental results show that our approaches outperform two strong baselines. We will release our code and data online.

In this paper, we focus on *graph-structured dataset construction under a new annotation guideline based on an existing source dataset and a small set of parallel annotated data*. Our contributions are summarized as follows.

- We propose the Label Switching conversion approach that generates pseudo target annotation via data augmentation.
- We propose the Graph2Graph Linear Transfer conversion approach that effectively transfer source graph information to target graph.
- We verify the effectiveness of the proposed approaches on five parallel annotated datasets.

2 Background

2.1 Semantic Dependency Graph

Chinese semantic dependency graph (SDG) (Che et al., 2016) is a framework for representing the meaning of different semantic units within a sentence (e.g., event chains, events, arguments, and concepts). It is in the form of directed acyclic graphs and focuses on investigating deeper semantic relations within sentences rather than morpho-syntactic patterns compared with traditional syntactic dependency trees. With the benefits of the graph's reentrancies and the easy-to-understand semantic labels, the tokens are connected more closely, making it easier to directly answer questions like *who did what to whom when and where*.

This framework is designed for Chinese exclusively. To take advantages of its properties, we modified the original annotation guidelines to make them applicable to English. We manually annotated 1,000 English sentences from UD-EWT to build a parallel annotated dataset to evaluate our annotation conversion approaches. Please refer to the Appendix for the modifications we made to the Chinese SDG guidelines.

2.2 Biaffine Graph Parser

In this paper, we build all the approaches over the state-of-the-art biaffine graph parser (Dozat and Manning, 2018), which is a graph-based dependency parser that employs biaffine classifiers to

[1] Referred to as UD-EWT in the rest of the paper.

predict arcs and labels in a graph. Firstly, it encodes the input sentence with a multi-layer bidirectional LSTM. Conventionally, the static word embeddings are used as the input vector. To exploit the capability of pretrained models in capturing structural information, we instead employ RoBERTa (Liu et al., 2019) to obtain the contextual representation as input. Secondly, the output of the LSTM of the i-th word, denoted as $\boldsymbol{h}_i$, is fed to four single-layer feed-forward networks (FFN) to get head and dependent representations for arcs (Eq. 1) and labels (Eq. 2).

$$\begin{aligned} \boldsymbol{h}_i^{(\text{arc-head})} &= \text{FFN}^{(\text{arc-head})}(\boldsymbol{h}_i) \\ \boldsymbol{h}_i^{(\text{arc-dep})} &= \text{FFN}^{(\text{arc-dep})}(\boldsymbol{h}_i) \end{aligned} \quad (1)$$

$$\begin{aligned} \boldsymbol{h}_i^{(\text{rel-head})} &= \text{FFN}^{(\text{rel-head})}(\boldsymbol{h}_i) \\ \boldsymbol{h}_i^{(\text{rel-dep})} &= \text{FFN}^{(\text{rel-dep})}(\boldsymbol{h}_i) \end{aligned} \quad (2)$$

Eventually, the scores for arcs (Eq. 4) and labels (Eq. 5) are computed with biaffine classifiers:

$$\text{Biaf}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \mathbf{U} \boldsymbol{x}_j + \boldsymbol{W}(\boldsymbol{x}_i \oplus \boldsymbol{x}_j) + \boldsymbol{b} \quad (3)$$

$$s_{i,j}^{(\text{arc})} = \text{Biaf}^{(\text{arc})}(\boldsymbol{h}_i^{(\text{arc-head})}, \boldsymbol{h}_j^{(\text{arc-dep})}) \quad (4)$$

$$\boldsymbol{s}_{i,j}^{(\text{rel})} = \text{Biaf}^{(\text{rel})}(\boldsymbol{h}_i^{(\text{rel-head})}, \boldsymbol{h}_j^{(\text{rel-dep})}) \quad (5)$$

For the labeled parser, $\mathbf{U} \in \mathbb{R}^{d \times c \times d}$, where c is the number of labels. While for the unlabeled parser, $\mathbf{U} \in \mathbb{R}^{d \times 1 \times d}$, so that $s_{i,j}^{(\text{arc})}$ is a scalar. The predictions of arcs and labels are $y_{i,j}'^{(\text{arc})} = \{s_{i,j} \geq 0\}$ and $y_{i,j}'^{(\text{rel})} = \text{argmax}\, \boldsymbol{s}_{i,j}$ respectively.

3 Method

In this section, we first give a formal definition of the task of supervised graph-to-graph annotation conversion (Section 3.1). Then, we present the proposed approaches, namely Label Switching (Section 3.2) and Graph2Graph Linear Transfer (Section 3.3) for this task.

3.1 Problem Definition

Given a set of texts $\mathcal{T}$, a graph-structured dataset annotated following guideline s on it is denoted by $D_s(\mathcal{T})$. In this paper, s is called the source guideline and $D_s(\mathcal{T})$ the source dataset. Assume we have a target guideline t as well as a small set of texts $\mathcal{T}' \subseteq \mathcal{T}$ annotated under t. In other words, we have the annotations of $\mathcal{T}'$ following both s (i.e., $D_s(\mathcal{T}')$) and t (i.e., $D_t(\mathcal{T}')$), which consist the parallel annotated dataset. The goal of supervised graph-to-graph annotation conversion is to learn a transformation $f : D_s(\mathcal{T}) \rightarrow D_t(\mathcal{T})$ based on $D_s(\mathcal{T}')$ and $D_t(\mathcal{T}')$, which converts the whole source dataset $D_s(\mathcal{T})$ into the target guideline, and thus get the annotated target dataset $D_t(\mathcal{T})$.

3.2 Label Switching

The lack of training data under the target guideline is a great challenge in supervised annotation conversion, especially for models based on deep neural networks. Data augmentation has been commonly used in the NLP community to alleviate the problem. Recently, Qin et al. (2020) proposed a code-switching data augmentation method, which generates pseudo multilingual corpus for the training of the multilingual BERT by randomly replacing words in a monolingual corpus based on bilingual dictionaries.

Inspired by this work, we propose the Label Switching approach that constructs pseudo target annotations to help the training of the conversion model by switching labels in source annotations to labels in the target guideline based on the alignment information obtained from the parallel annotated data. Our Label Switching approach consists of two steps: (i) label-switching data augmentation and (ii) two-step fine-tuning, which are introduced as follows.

Label-Switching Data Augmentation: To construct pseudo training data under the target guideline, we first compute the label alignment-based switching probabilities on the parallel annotations $D_s(\mathcal{T}')$ and $D_t(\mathcal{T}')$. Specifically, for a text $X \in T'$, its source and target annotations are denoted by $D_s(X)$ and $D_t(X)$ respectively. Let (i, j, r) denote the arc from word i to word j with label r, we count the number of the quadruples (r_t, p_h, p_d, r_s) for all the arcs that exist in both source and target annotations (i.e., $(i, j, r_s) \in D_s(X)$ and $(i, j, r_t) \in D_t(X)$), where p_h and p_d are the Part-of-Speech (POS) tags for the head and dependent words respectively.[2] The switching probability is thus computed as:

$$P(r_t | p_h, p_d, r_s) = \frac{\text{N}_{(r_t, p_h, p_d, r_s)}}{\sum_{r' \in \mathcal{R}_t} \text{N}_{(r', p_h, p_d, r_s)}}, \quad (6)$$

where $\text{N}_{(r_t, p_h, p_d, r_s)}$ is the number of the quadruples in the parallel annotated data, and $\mathcal{R}_t$ is the set of

[2]We use gold POS tags from the source dataset in our experiments.

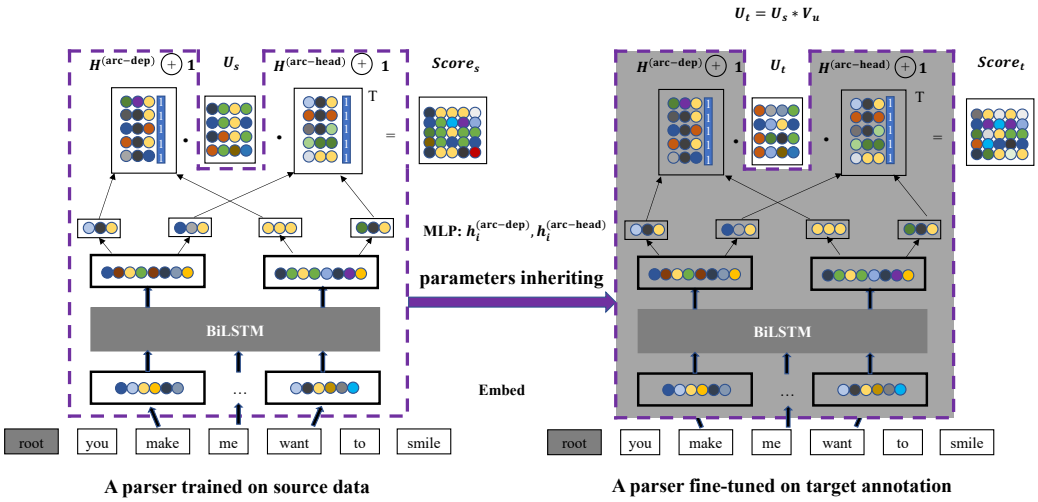

Figure 2: Schematic diagram of Graph2Graph Linear Transfer which scores each possible head for each dependent. **Embed** refers to embedding layer. The parameters of a biaffine parser trained on source data is inherited by another parser with a linear transfer function applied to its biaffine attention matrix. All the inherited parameters are fixed while only the linear transfer function is fine-tuned on target data.

all the labels in the target guideline. Eventually, each label r_s in the source dataset, with POS tags p_h and p_d for the head and dependent respectively, is switched to r_t in the target guideline with the probability $P(r_t|p_h, p_d, r_s)$.[3]

Two-Step Fine-Tuning: The pseudo target data generated in the last step is firstly used to fine-tune a pretrained model-based biaffine graph parser as described in Section 2.2. Secondly, the model is further fine-tuned on the manually annotated target data $D_t(\mathcal{T}')$. Eventually, this parser is used to generate the annotation of the whole dataset under the target guideline with only texts as input.

3.3 Graph2Graph Linear Transfer

Compared with the Label Switching approach which transforms the annotations through data augmentation and two-step fine-tuning, our second approach directly learns a linear function that transforms the parser trained on the source data to the one that fit the target annotation guideline. Since the biaffine attention matrix is the core of the biaffine parser and contains knowledge that is significant for the prediction of the dependency graph. A natural way to exploit source graph information is to inherit such knowledge from a parser trained on the source data.

As illustrated in Figure 2, to exploit the annotation information under source guideline, we utilize the knowledge learned in a biaffine parser including pretrained model parameters, head and dependent representations for arcs and labels, which is trained on the source dataset. In order to be adapted to target annotation, a linear transfer function is designed for learning target biaffine attention matrix.

Specifically, let $\mathbf{U}_s$, $\boldsymbol{W}_s$ and $\boldsymbol{b}_s$ be the parameters of a biaffine parser trained on the source dataset with Eq. 3. Two linear transfer functions $\boldsymbol{V}_u$ and $\boldsymbol{V}_w$ are applied to $\mathbf{U}_s$ and $\boldsymbol{W}_s$ respectively to obtain the parameters $\mathbf{U}_t$ and $\boldsymbol{W}_t$ for the target parser.

$$\mathbf{U}_t = \mathbf{U}_s * \boldsymbol{V}_u \tag{7}$$

$$\boldsymbol{W}_t = \boldsymbol{W}_s * \boldsymbol{V}_w \tag{8}$$

$$\text{Biaf}_t(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \mathbf{U}_t \boldsymbol{x}_j + \boldsymbol{W}_t(\boldsymbol{x}_i \oplus \boldsymbol{x}_j) + \boldsymbol{b}_s \tag{9}$$

Our Graph2Graph Linear Transfer approach is divided into two steps: i) training a biaffine parser on the source data; ii) employing a biaffine parser with linear transfer function applied to its attention matrix to inherit the parameters obtained in step i), then fine-tuning the new parser on the target data while freezing the inherited parameters.

4 Experimental Setup

4.1 Datasets and Experimental Settings

Recall that this paper aims to construct a dataset under a new target guideline based on the existing

[3]Due to the limited number of parallel annotated data, the switching probabilities can not cover all the labels in the source data. For those not covered, we leave them as they are.

Dataset	#Sent	#Token	#Arc (avg)	#Label
UD-EWT	16,622	254,829	1.00	49
SDG	1,000	15,991	1.07	71
UD-En.	16,622	254,829	1.05	398
DM	38,915	866,248	0.78	61
PAS	38,915	866,248	1.02	43
PSD	38,915	866,248	0.70	92

Table 1: Data statistics. **#Sent** and **#Token** denote the number of sentences and tokens respectively for all the annotated data in the dataset (including training, valid and test sets). **#Arc (avg)** denotes the average number of arcs per token, while **#Label** the number of label types. **UD-En.** denotes the UD-Enhanced dataset.

source dataset and a small set of parallel annotated data. For the evaluation of the proposed approaches, we manually construct the SDG dataset (on part of texts from UD-EWT) and employ two existing parallel annotated datasets, namely {UD-EWT, UD-Enhanced} and {DM, PAS, PSD}, whose statistics are shown in Table 1.

UD-EWT is a tree-structured syntactic dataset under the Universal Dependencies (UD) guideline. UD (Zeman et al., 2019) is a framework for consistent annotation of grammar across languages. The UD Treebanks (v2.5)[4] consist of 157 treebanks in 90 languages, which could be a good source to obtain source datasets for dataset construction under a new guideline. Therefore, we use UD-EWT as the source dataset in our experiments.

UD-Enhanced is a graph-structured syntactic dataset converted from UD-EWT by adding relations and augmenting relation names (Schuster and Manning, 2016).[5]

SDG is a graph-structured semantic dataset with 1,000 sentences annotated under the refined semantic dependency graph guideline (Che et al., 2016).

DM, PAS and PSD are three types of graph-structured semantic annotations in the SemEval 2015 task 18 dataset (Oepen et al., 2015).[6]

The approaches are evaluated on the following five annotation conversion tasks. Note that besides the parallel annotated data, the whole source dataset is available in the training process.

UD-EWT to UD-Enhanced We randomly select 1,000/500/5,000 parallel annotated sentences from the two datasets as training/valid/test sets.

UD-EWT to SDG We perform 5-fold cross-validation on the 1,000 parallel annotated sentences.

DM to PAS / PAS to DM / PAS to PSD We randomly select 1,000/500/10,000 parallel annotated sentences from the two datasets as training/valid/test sets.

For all the approaches, we employ the biaffine graph parser as described in Section 2.2 to predict the target graph, and use RoBERTa (Liu et al., 2019) to obtain contextual representations as its input. We set the learning rate of RoBERTa to 2e-5 and that of other parameters to 2e-2. Other hyperparameters are adopted from the paper of Dozat and Manning (2018).

4.2 Baselines and Evaluation Metrics

Our approaches, namely **Label Switching** and **Graph2Graph Linear Transfer**, are compared with two RoBERTa-based strong baselines, introduced as follows.

Direct Fine-Tuning (DFT) A RoBERTa-based biaffine graph parser is directly fine-tuned on the small set of target annotations $D_t(\mathcal{T}')$.

Two-Step Fine-Tuning (TSFT) A RoBERTa-based biaffine graph parser is first fine-tuned on the whole source dataset $D_s(\mathcal{T})$, and then further fine-tuned on the small set of target annotations $D_t(\mathcal{T}')$.[7] It is only fine-tuned for 5 epochs in the first step to avoid over-fitting to the source data.

Moreover, It is straightforward to combine the two approaches we proposed **(LS+GGLT)** by averaging the scores they predicted for arcs (Eq. 4) and labels (Eq. 5) respectively. This is also evaluated in the experiments.

All the results are reported in terms of unlabelled precision (UP), recall (UR) and F1 score (UF) and labelled precision (LP), recall (LR) and F1 score (LF) on the target test set.

5 Results

Model	UP	UR	UF	LP	LR	LF
DFT	89.76	90.74	90.25	86.37	87.31	86.83
TSFT	91.51	93.16	92.33	88.29	89.89	89.08
LS	97.81	98.23	98.02	96.36	96.79	96.57
GGLT	97.84	98.15	97.99	96.32	96.63	96.48
LS+GGLT	**98.14**	**98.28**	**98.21**	**96.80**	**96.94**	**96.87**

Table 2: Results for conversion from UD-EWT to UD-Enhanced.

[4] http://hdl.handle.net/11234/1-3105

[5] https://github.com/UniversalDependencies/UD_English-EWT

[6] https://catalog.ldc.upenn.edu/LDC2016T10

[7] In the second step, non-RoBERTa parameters are reinitialized since the source and target guidelines have different label sets, and thus only the RoBERTa parameters can be shared.

Table 2 shows the results for conversion from UD-EWT to UD-Enhanced. With the help of the pretrained model, DFT achieves fair results with only 1,000 sentences annotated under the target guideline for training. While the other baseline, TSFT, improves the results by fine-tuning on the large scale source dataset first to capture the structural information implicitly. As for our approaches, both of them significantly outperform the baselines. With only implicit parameterized information used during the training period, GGLT yields results comparable to LS which exploits large scale pseudo target data switched from the source dataset. Furthermore, the combined approach achieves a better result. The gains are 0.30 (96.87 - 96.57) in LF compared to LS and 0.39 (96.87 - 96.48) compared to GGLT.

Model	UP	UR	UF	LP	LR	LF
DFT	85.40	86.01	85.70	73.29	73.79	73.53
TSFT	87.26	87.78	87.51	74.44	74.86	74.64
LS	89.73	89.57	89.64	77.84	77.67	77.75
GGLT	89.45	**89.94**	89.69	77.42	77.83	77.62
LS+GGLT	**90.52**	89.87	**90.19**	**78.63**	**78.05**	**78.33**

Table 3: Results (averaged across 5-fold cross-validation) for conversion from UD-EWT to SDG.

Table 3 shows the results for conversion from UD-EWT to SDG, which are averaged across 5-fold cross-validation. For baselines, TSFT achieves higher results than DFT since it uses large scale source data during training while DFT does not. For our approaches, both LS and GGLT outperform the baselines. Similar to the situation in conversion from UD-EWT to UD-Enhanced, LS and G2G achieve similar results. Moreover, the combined approach achieves the best results, which are higher than that of both LS and GGLT.

Model	UP	UR	UF	LP	LR	LF
DFT	93.50	93.98	93.74	91.30	91.76	91.53
TSFT	93.72	93.95	93.84	91.55	91.78	91.67
LS	94.52	94.79	94.65	92.66	92.92	92.79
GGLT	94.34	**95.00**	94.67	92.35	92.99	92.67
LS+GGLT	**94.80**	94.89	**94.85**	**92.98**	**93.07**	**93.02**

Table 4: Results for conversion from DM to PAS.

The results for conversion from DM to PAS, PAS to DM and PAS to PSD are shown in Table 4, 5 and 6 respectively. For the two baseline models, in all cases, TSFT performs better than DFT. As for our approaches, LS and GGLT consistently outperform the two baselines. Also, the combined approach achieves the best results in all cases including conversion from PAS to PSD, conversion from PAS to DM and conversion from DM to PAS.

Model	UP	UR	UF	LP	LR	LF
DFT	89.46	89.45	89.46	86.81	86.80	86.80
TSFT	90.19	89.74	89.96	87.50	87.07	87.28
LS	90.74	**91.30**	91.02	88.28	88.82	88.55
GGLT	90.60	**91.30**	90.95	88.17	88.84	88.50
LS+GGLT	**91.32**	91.27	**91.30**	**88.98**	**88.94**	**88.96**

Table 5: Results for conversion from PAS to DM.

Model	UP	UR	UF	LP	LR	LF
DFT	90.19	91.59	90.89	74.79	75.95	75.37
TSFT	90.72	92.40	91.56	74.86	76.25	75.55
LS	92.69	**93.64**	93.08	77.30	77.94	77.62
GGLT	91.97	93.44	92.70	76.15	77.37	76.75
LS+GGLT	**93.11**	93.46	**93.28**	**78.17**	**78.46**	**78.31**

Table 6: Results for conversion from PAS to PSD.

6 Discussion

6.1 Effect of Dataset Similarity

Intuitively, it is believed that the more similar the source guideline is to the target one, the easier it will be to convert an existing source dataset into the target guideline. The similarity between the two guidelines can also be understood as the amount of shared information that can be used to convert annotations from one to the other. However, it is hard to measure the similarity between the two annotation guidelines. Therefore, we instead compute the similarity between parallel annotated datasets and explore its effect on the annotation conversion. Specifically, for each of the five conversion tasks introduced in Section 4.1, we directly evaluate the original source dataset on the gold target dataset and use the results to measure the similarity between the two datasets.

Source	Target	UF	LF
UD-EWT	**UD-Enhanced**	96.84	83.30
UD-EWT	**SDG**	86.73	-
DM	**PAS**	64.54	-
PAS	**DM**	64.54	-
PAS	**PSD**	27.15	-

Table 7: Dataset similarities in terms of UF and LF.

Results are shown in Table 7, we only report the LF for the dataset pair of {UD-EWT, UD-Enhanced}. This is because UD-Enhanced is converted from UD-EWT by adding relations and augmenting relation names to make implicit relations between content words more explicit. Therefore, UD-Enhanced shares some labels with UD-EWT.

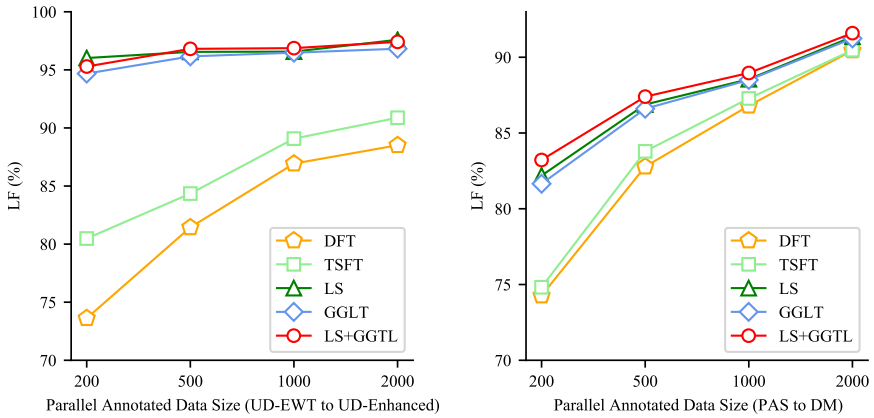

Figure 3: Results for conversion from UD-EWT to UD-Enhanced and PAS to DM with different parallel annotated data sizes (best viewed in color).

While in all the other pairs, the two datasets have completely different label sets. Thus we can not compute the LF for them.

As for the UF which reflects the structural similarity between datasets, we find that UD-EWT is most similar to UD-Enhanced, which can also be explained by the construction of UD-Enhanced introduced above. The similarity between UD-EWT and SDG is lower, indicating that the conversion from UD-EWT to SDG is harder than that from UD-EWT to UD-Enhanced. Moreover, the similarities between DM, PAS and PSD are much lower, with only 27.15% UF for PAS and PSD, which suggests that the shared information between them is much less than that for the other pairs and conversion between them are even more challenging. According to results reported in Section 5, the more similar the source and target datasets are, the higher improvements our proposed approaches can obtain.

6.2 Effect of Parallel Annotated Data Size

Recall that this paper aims to construct a dataset under a new annotation guideline based on the existing source dataset and little human labour. Therefore, the parallel annotated data size is of great importance since the smaller it is, the less human labour will be required. This section investigates the effect of the parallel annotated data size on the proposed approaches in the conversion from UD-EWT to UD-Enhanced and conversion from PAS to DM. Specifically, we evaluate the approaches on 200/500/1,000/2,000 randomly selected training sets respectively with the same valid/test sets introduced in Section 4.1.

Figure 3 shows the results with different parallel annotated data sizes, where it is obvious that the performances of all methods increase as the data size increases in the annotation conversion from PAS to DM. It suggests that the more annotated data is employed, the better result we will get. However, the performance of LS, GGLT and the combined approach is not apparently influenced by the change of data size in the annotation conversion from UD-EWT to UD-Enhanced. It can be explained by the similarity between them. With high similarity, our proposed approaches can easily obtain promising results with only 200 parallel annotated sentences. Another finding from Figure 3 is that the difference of LF between our proposed approaches and the baselines shrinks as the data size increases, which may indicates that our proposed approaches are most suitable for cases where only limited parallel annotated data is available. And this is exactly the aim of this paper.

6.3 Quality of Pseudo Target Data

The proposed Label Switching approach performs consistently well across all the five annotation conversion tasks in Section 5. In this section, we explore its property by analyzing the quality of pseudo target data generated in the core data augmentation step of the approach. Specifically, we directly evaluate the pseudo target dataset switched from the source dataset on the gold target dataset to measure their quality.

Table 8 shows the qualities of the pseudo target data in terms of UF and LF on five annotation conversion tasks. Firstly, it is obvious that the UF of each task is identical to those in Table 7. This is because only labels in the source dataset are switched

Source	Target	UF	LF
UD-EWT	**UD-En.**	96.84	86.46
UD-EWT	**SDG**	86.73	58.52
DM	**PAS**	64.54	59.85
PAS	**DM**	64.54	57.91
PAS	**PSD**	27.15	21.53

Table 8: Qualities of the pseudo target data in terms of UF and LF. **UD-En.** denotes UD-Enhanced.

in the data augmentation step, while the arcs are not changed.

As described in Section 3.2, due to the limited number of parallel annotated data, the switching probabilities computed from it can not cover all the labels in the source data. Most of the source labels are covered in the data augmentation step on conversion from UD-EWT to UD-Enhanced and SDG. However, the quality of the pseudo UD-Enhanced data is much higher than that of the pseudo SDG data. This is because UD-Enhanced shares parts of its label set with UD-EWT, while SDG has a completely different label set. As for DM, PAS, and PSD, only half to three-fourths of the source labels are covered, and the qualities of the pseudo data are even lower. However, despite the low-quality pseudo data, the approach is still effective and outperforms the two strong baselines.

	UD-EWT2UD-EN	PAS2DM
Data Size	**LF**	**LF**
200	86.41	55.94
500	86.28	57.11
1,000	86.46	57.91
2,000	86.54	58.14

Table 9: Qualities of the pseudo target data on conversion from UD-EWT to UD-Enhanced and PAS to DM with different parallel annotated data sizes.

In Section 6.2, we find that whether the performance of the Label Switching approach is influenced by the parallel annotated data size seems to depend on conversion itself. To explore the underlying reason, we compute the qualities of the pseudo target data and the coverage rate of switched source labels on conversion from UD-EWT to UD-Enhanced and PAS to DM with different parallel annotated data sizes. Results in Table 9 show that the labelled score on conversion from PAS to DM increases as the size increases, while the score remains almost the same on conversion from UD-EWT to UD-Enhanced. This further verifies the correlation between the quality of pseudo target data and the approach's performance, and may to some extent explain why its performance does not change with the parallel annotated data size on conversion from UD-EWT to UD-Enhanced. So the quality of the pseudo target data contributes to the performance of Label Switching method.[8]

7 Conclusion

This paper aims at graph-structured dataset construction under a new annotation guideline based on an existing dataset with little human labour. We propose two pretrained model-based graph-to-graph annotation conversion approaches, namely Label Switching and Graph2Graph Linear Transfer, and show their effectiveness on five annotation conversion tasks. Results show that 1) the Label Switching approach and Graph2Graph Linear Transfer perform consistently well across all the tasks; 2) our proposed methods are suitable for cases where only limited parallel annotated data is available; 3) the two approaches can be combined to further improve the performance.

8 Ethical Considerations

The sentences in the Semantic Dependency Graph (SDG) dataset we construct are collected from the English Web Treebank (EWT) in the Universal Dependencies (UD) Treebanks (v2.5) (Zeman et al., 2019) which is a publicly available dataset. The detailed statistics of the SDG dataset are shown in Table 1. All the annotators are voluntary participants who have given informed consent and been fairly compensated during the annotation process.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Wanxiang Che, Yanqiu Shao, Ting Liu, and Yu Ding. 2016. SemEval-2016 task 9: Chinese semantic dependency parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1074–1080, San Diego, California. Association for Computational Linguistics.

[8]Note that the sizes of pseudo data used in the first fine-tuning step are the same for different parallel annotated data sizes.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Wenbin Jiang, Yajuan Lü, Liang Huang, and Qun Liu. 2015. Automatic adaptation of annotations. *Computational Linguistics*, 41(1):119–147.

Xinzhou Jiang, Zhenghua Li, Bo Zhang, Min Zhang, Sheng Li, and Luo Si. 2018. Supervised treebank conversion: Data and approaches. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2706–2716, Melbourne, Australia. Association for Computational Linguistics.

Xiang Li, Wenbin Jiang, Yajuan Lü, and Qun Liu. 2013. Iterative transformation of annotation guidelines for constituency parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–596, Sofia, Bulgaria. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics.

Stephan Oepen, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado. Association for Computational Linguistics.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3853–3860. International Joint Conferences on Artificial Intelligence Organization. Main track.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378, Portorož, Slovenia. European Language Resources Association (ELRA).

Daniel Zeman, Joakim Nivre, Mitchell Abrams, and et al. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Muhua Zhu, Jingbo Zhu, and Minghan Hu. 2011. Better automatic treebank conversion using a feature-based approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 715–719, Portland, Oregon, USA. Association for Computational Linguistics.

A Appendix

A.1 Semantic Dependency Graph Annotation Guidelines

We modified Chinese Semantic Dependency Graph guidelines[9] to make it applicable to English in two ways: adding more semantic edges and reducing more semantic labels.

We added more edges between predicates and arguments. In the Chinese Semantic Dependency Graph, it only considers omitted object and subject which has been referred in previous clauses. We also take omitted predicates into account, thus, ensuring the semantic integrity of semantic units. An example is shown in Figure 4. Here, the predicate "cried" has been omitted and we added an extra edge to connect "I" with "cried" which makes the second clause more explicit.

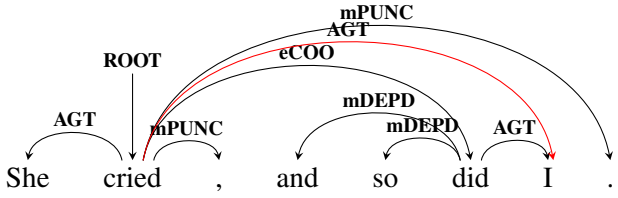

Figure 4: Example of annotation for omitted predicates

As for semantic labels, we merged labels for simplification. Specifically, in semantic roles, we merged Aft into EXP, Orig and Comp into DATV, Reas, Int into REAS, Host, Nmod, Tmod into FEAT, Qp, Freq, Seq into QUAN. In event relations, we merged eInf, eCau into eRESU, eConc and eAban into eSELT, eSUM into eRECT. In semantic markers, we just kept mNEG, mRELA and mPUNC and abandoned other markers because they most designed for Chinese specifically. We also create some new labels for unique usage in English: mFIXED for multi-word expressions(mwe) and mDEPD for function words like articles. The list of the semantic labels in our SDG guideline is shown in Table 10.

Semantic Class	**Labels**
Semantic roles	AGT(Agent), EXP(Experiencer), PAT(Patient), CONT(Content), PROD(Product), BELGONG(Belongings), PART, MATL(Material), TOOL, REAS(Reason), LOC(location), TIME, SCO(Scope), FEAT, QUAN(Quantity), STAT(State)
Reverse relations	r+semantic roles
Nested relations	d+semantic roles
Event relations	eCOO(Coordination), eRECT(Recount), eSELT(Select), ePROG(Progression), eSUCC(Successor), eRESU(Result), eCOND(Condition), eSUPP(Supposition), eEFTT(Effect), eEQU(Equal), eADVT(adversative)
Semantic markers	mNEG(Negation), mRELA(relation), mPUNC(Punctuation), mDEPD, mFIXED

Table 10: Label set of the semantic relation of EN-SDG

[9] https://csdp-doc.readthedocs.io/zh_CN/latest/