Simple and Effective Graph-to-Graph Annotation Conversion

Yuxuan Wang[♦][*], Zhilin Lei[♡][*], Yuqiu Ji[♡], Wanxiang Che[♡][†]
[♦]Zhejiang Lab
[♡]Research Center for Social Computing and Information Retrieval,
Harbin Institute of Technology
yxwang@zhejianglab.com, {zllei, yqji, car}@ir.hit.edu.cn

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 conversions. In this paper, we propose two simple and effective graph-to-graph annotation conversion approaches, namely Label Switching and Graph2Graph Linear Transformation, which use pseudo data and inherit parameters to guide graph conversions respectively. These methods are able to deal with conversion between graph-structured annotations and require no manually designed features. To verify their effectiveness, we manually construct a graph-structured parallel annotated dataset and evaluate the proposed approaches on it as well as other existing parallel annotated datasets. Experimental results show that the proposed approaches outperform strong baselines with higher *conversion score*. To further validate the quality of converted graphs, we utilize them to train the target parser and find graphs generated by our approaches lead to higher *parsing score* than those generated by the baselines.[1]

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. Some others, such as the Enhanced English Universal Dependencies dataset (Schuster and Manning, 2016), are converted from existing dataset 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.

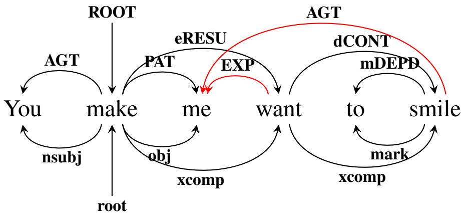

Figure 1: Example of annotation conversion from Universal Dependency Tree (bottom) to our Semantic Dependency Graph (top).

Obviously, in the dataset construction process under a new annotation guideline, it would be extremely expensive to annotate the whole dataset manually. Although rule-based conversion needs no human labour for annotation, it 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 datasets and learn a transduction that converts them into a 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.

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; Seddah et al., 2018). In their cases, two existing heterogeneous treebanks (tree-structured datasets) on different texts are available. The goal is to convert a source treebank into annotation under a target guideline and use the converted treebank as extra annotated data for the training of the target model.

[*]Equal Contribution.
[†]Email Corresponding.
[1]Our code and dataset are available at `https://github.com/WangYuxuan93/G2GConversion`.

Previous work for treebank conversion mainly focused on rule-based and feature-based methods. For the rule-based methods (Frank, 2001; Schuster and Manning, 2016), treebank-specific rules are designed by experts to convert one treebank to the other one. For the feature-based methods, 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 annotation as extra guiding features to train an augmented target parser that parses the whole source treebank and generates the expected target annotation. Such methods are not easily applicable to new conversions, especially between graph-structured representations, since the annotation guidelines are normally vastly different from each other, and thus the rules or features should be redesigned for every new guideline.

In this paper, we propose two pretrained model-based graph-to-graph annotation conversion approaches, namely Label Switching (LS) and Graph2Graph Linear Transformation (G2GLT), which are able to deal with conversions between graph-structured datasets and require no manually designed features. 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 graph parser is first trained on the pseudo data and then further fine-tuned on the small set of gold target annotation. The parser is eventually used to parse the source dataset to generate the target annotation. While the G2GLT approach directly inherits most of parameters from the parser trained on the source annotation, then linearly transform the biaffine attention matrix in a biaffine graph parser (Dozat and Manning, 2018) to adapt to the target annotation guideline.

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) (Silveira et al., 2014) in the Universal Dependencies (UD) Treebanks (v2.5) (Zeman et al., 2019).[2] To verify the effectiveness of the proposed approaches, we evaluate them on conversions between 6 datasets including SDG, UD-EWT, the Enhanced Universal Dependencies (UD-Enhanced) dataset (Schuster and Manning, 2016) and three types of annotations (i.e., DM, PAS and PSD) in the SemEval 2015 task 18 dataset (Oepen et al., 2015). We further validate the quality of the converted annotations by utilizing them in the training of the target parser. Experimental results show that our approaches outperform strong baselines on both *conversion score* and *parsing score*.

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

- We propose two graph-to-graph conversion approaches that require no manually designed features.
- We verify the effectiveness of our proposed approaches on 5 existing datasets and a graph-structured dataset manually constructed by ourselves.
- We validate the quality of the annotations converted by our approaches by utilizing them to train the target parser.

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 Appendix A.2 for the modifications we made to the Chinese SDG guidelines.

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

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 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).

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

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

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

$$\text{Biaf}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \mathbf{U} \boldsymbol{x}_j + \boldsymbol{W}([\boldsymbol{x}_i; \boldsymbol{x}_j]) + \boldsymbol{b} \tag{3}$$

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

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

Where $[\boldsymbol{x}_i; \boldsymbol{x}_j]$ indicates the concatenation of the two vectors. For the labeled parser, $\mathbf{U} \in \mathbb{R}^{d \times c \times d}$ and $\boldsymbol{W} \in \mathbb{R}^{c \times 2d}$ where c is the number of relation labels and d is the dimension of hidden states. While for the unlabeled parser, $\mathbf{U} \in \mathbb{R}^{d \times 1 \times d}$ and $\boldsymbol{W} \in \mathbb{R}^{1 \times 2d}$, so that $s_{i,j}^{(\text{arc})}$ is a scalar. The predictions of arcs and labels are $y_{i,j}^{\prime(\text{arc})} = \{s_{i,j}^{(\text{arc})} \geq 0\}$ and $y_{i,j}^{\prime(\text{rel})} = \arg\max \boldsymbol{s}_{i,j}^{(\text{rel})}$ respectively. Where the latter means that the label with the highest score is the prediction.

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 Transformation (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 data. 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 annotation under the target guideline, and thus obtain 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-

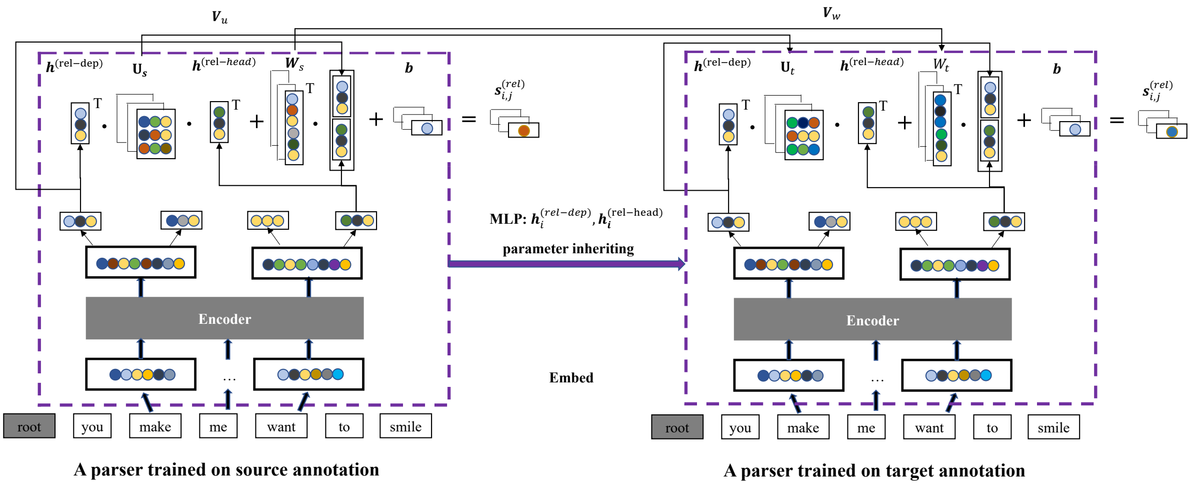

Figure 2: Schematic diagram of Graph2Graph Linear Transformation which scores each relation label between the head and the dependent. **Embed** refers to embedding layer. The parameters of a biaffine parser trained on source data is inherited by another parser with a linear transformation function applied to its biaffine attention matrix. All the inherited parameters and the linear transformation function are fine-tuned on target data except for the biaffine attention matrix inherited from the source parser.

of-Speech (POS) tags for the head and dependent words respectively.[3] The switching probability is thus computed as:

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

where $\mathrm{N}_{(r_t,p_h,p_d,r_s)}$ is the number of the quadruples in the parallel annotated data, and $\mathcal{R}_t$ is the set of 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 $P(r_t|p_h, p_d, r_s)$.[4]

Two-Step Fine-Tuning: The pseudo target data generated in the last step is firstly used to train a 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 Transformation

Compared with the Label Switching approach which converts the annotation through data augmentation and two-step fine-tuning, our second approach directly learns a linear function that transforms the parser trained on the source-annotated data to a parser that fits 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 large-scale source-annotated data.

As illustrated in Figure 2, to exploit the information in the source data, we firstly train a source parser on it. Then a linear transformation is applied to the biaffine attention matrix for relation prediction so that the relational information learned on the source annotation can be transformed to the target annotation. To maintain the source relational information, the corresponding biaffine attention matrix is fixed during the fine-tuning stage on the target data.

Specifically, let $\mathbf{U}_s$, $\boldsymbol{W}_s$ and $\boldsymbol{b}_s$ be the parameters of a biaffine parser trained on large-scale source-annotated data with Eq. 3. Let c_s and c_t be the number of relation labels in the source and target annotations respectively. Two linear transformation functions $\boldsymbol{V}_u \in \mathbb{R}^{c_t \times c_s}$ and $\boldsymbol{V}_w \in \mathbb{R}^{c_t \times c_s}$ are applied to $\mathbf{U}_s \in \mathbb{R}^{d \times c_s \times d}$ and $\boldsymbol{W}_s \in \mathbb{R}^{c_s \times 2d}$ respectively to obtain the parameters $\mathbf{U}_t \in \mathbb{R}^{d \times c_t \times d}$ and $\boldsymbol{W}_t \in \mathbb{R}^{c_t \times 2d}$ for the target parser.

$$\mathbf{U}_t = (\boldsymbol{V}_u \mathbf{U}_s^{\top(1,2)})^{\top(1,2)} \quad (7)$$

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

[4] 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.

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

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

where $\mathbf{U}^{\top(1,2)}$ means transposing the first and the second dimensions of tensor $\mathbf{U}$.

Briefly, our Graph2Graph Linear Transformation approach consists of two steps: (i) training a source biaffine parser on the source-annotated data; (ii) applying the linear transformation function on the source parser and fine-tuning it on the target-annotated data while freezing the parameters in the biaffine attention matrix inherited from the source parser.

4 Experimental Setup

4.1 Datasets and Experimental Settings

Dataset	#Sent	#Token	#Arc (avg)	#Label
UD-EWT	16,622	254,829	1.00	49
SDG	1,000	15,991	1.07	61
UD-En	16,622	254,829	1.05	398
DM	37,066	834,665	0.78	60
PAS	37,066	834,665	1.02	43
PSD	37,066	834,665	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. For **DM/PAS/PSD**, the out-of-domain test sets are excluded.

Dataset	Source-Train	Train	Valid	Test
UD2UD-En	10,508	1000	500	5,000
UD2SDG	16,369	800	-	200
D2D*	26,206	1,000	500	10,000

Table 2: Data split statistics. **UD** denotes UD-EWT. **D2D*** denotes the 6 conversion tasks between **DM**, **PAS** and **PSD**.

For the evaluation of the proposed approaches, we manually construct the SDG dataset (on part of texts from UD-EWT) and employ two groups of 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) consist of 157 treebanks in 90 languages,[5] 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).[6]

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).[7]

The approaches are evaluated on eight annotation conversion tasks including UD-EWT to UD-Enhanced (**UD2UD-En**), UD-EWT to SDG (**UD2SDG**) as well as six conversion tasks between DM, PAS and PSD (**PAS2DM**, **PSD2DM**, **PAS2PSD**, **DM2PSD**, **DM2PAS**, **PSD2PAS**). Recall that the goal of this paper is graph-structured annotation conversion based on an existing source-annotated dataset and a small set of parallel annotated data. To fit the goal, we re-split the datasets so that only a limited number of parallel annotated examples are available while training. The data split statistics are shown in Table 2, where **Train/Valid/Test** are parallel annotated data and **Source-Train** contains only the source-side annotation that will be used in the experiment of utilizing converted data in Section 5.2. We perform 5-fold cross-validation on the 1,000 parallel annotated sentences in the conversion task from UD-EWT to SDG.

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).

Existing graph banks can be broadly categorized into two types, namely the *bilexical graphs* (where dependencies are directly linked between surface lexical units) and the *conceptual graphs* (where

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

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

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

	UD2UD-En		UD2SDG		PAS2DM		PSD2DM		DM2PAS		PSD2PAS		DM2PSD		PAS2PSD		AVG.	
Methods	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF
DFT	89.92	82.04	86.47	74.07	90.07	87.53	90.07	87.53	93.76	91.56	93.76	91.56	90.87	75.32	90.87	75.32	90.72	83.12
TSFT	91.74	83.32	87.49	74.87	90.51	88.01	90.48	88.10	94.11	91.96	94.46	92.37	91.26	75.32	91.66	75.57	91.46	83.69
PE	**99.41**	**95.82**	87.68	74.93	89.79	87.33	90.11	87.68	93.93	91.75	94.20	91.90	89.69	74.03	91.81	76.54	92.08	85.00
G2GTr	96.63	89.16	87.85	74.91	90.11	87.62	90.58	88.20	93.81	91.63	94.42	92.28	90.53	74.21	91.88	75.73	91.98	84.22
LS	97.13	89.27	89.15	76.65	91.12	88.71	91.05	88.71	94.72	92.90	95.23	93.29	92.08	76.90	93.14	77.65	92.95	85.51
G2GLT	96.51	90.97	89.30	77.05	90.86	88.46	91.27	89.31	94.37	92.31	94.88	92.92	90.81	75.12	92.63	76.81	92.58	85.37
LS+G2GLT	97.27	91.47	**89.81**	**77.81**	**91.33**	**89.05**	**91.87**	**90.05**	**94.86**	**93.01**	**95.80**	**94.10**	**92.77**	**79.36**	**93.29**	**78.42**	**93.38**	**86.66**

Table 3: Conversion scores on test data.

dependencies are between virtual nodes that do not need be explicitly mapped to surface linguistic forms). DM, PAS and PSD fall into the former category, while another popular graph bank Abstract Meaning Representation (AMR) (Banarescu et al., 2013) belongs to the latter. We choose to conduct experiments on DM, PAS and PSD because they are annotated on the same corpus, which guarantees the availability of parallel annotated data required by our approaches. While currently we have no access to parallel annotated data for AMR and any other graph bank. More importantly, since our Label Switching approach exploits the overlapped dependencies in source and target annotation for label switching, it is not straightforwardly applicable to conversion between conceptual graphs since virtual nodes make it hard to identify overlapped dependencies. Therefore, we leave as future work to extend our approaches to conversion between conceptual graphs.

4.2 Baselines and Evaluation Metrics

Our approaches, namely **Label Switching (LS)** and **Graph2Graph Linear Transformation (G2GLT)**, are compared with four baselines introduced as follows.

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

Two-Step Fine-Tuning (TSFT) A RoBERTa-based biaffine graph parser is firstly trained on the whole source dataset $D_s(\mathcal{T})$, and then further fine-tuned on the small set of target annotations $D_t(\mathcal{T}')$.[8] It is only trained for 5 epochs in the first step to avoid over-fitting to the source data.

Pattern Embedding (PE) This is a feature-based method closely following the work of Jiang et al. (2018) which takes advantage of source guiding features. To adopt it in graph-to-graph annotation conversion, we average the label embeddings for structural information representation in the reentrancy structure and add reverse sibling pattern for the reentrancy structure in graphs.[9] A RoBERTa-based biaffine graph parser is employed in the method.

Graph2Graph Transformer (G2GTr) Graph2Graph Transformer is proposed by Mohammadshahi and Henderson (2021) for dependency parsing with iterative refinement, which encodes dependency trees produced by the last step to obtain structural enhanced representation that is utilized to predict refined trees. In this method, we employ Graph2Graph Transformer to obtain each token's representation infused with source annotation information and feed them to a RoBERTa-based biaffine graph parser that predicts the target annotation.

Moreover, it is straightforward to combine the two approaches we proposed **(LS+G2GLT)** 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 F1 score (UF) and labelled F1 score (LF) on the target test set.

5 Results

5.1 Conversion Results

Table 3 shows the results of the eight conversion tasks, which are the average over three runs.[10] With the help of the pretrained model, DFT achieves fair results with limited data annotated under the target guideline used for training. While TSFT improves the results by training on the large-scale source-annotated dataset firstly to capture the structural information implicitly. The other two baselines, which adopt previous methods for graph-to-graph annotation conversion, yield better results. Specif-

[8] 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.

[9] Reentrancy structure represent the structure of a word with multiple heads, which only occurs in graphs but not in trees.

[10] Please refer to Appendix A.3 for the standard deviation.

	UD2UD-En		UD2SDG		PAS2DM		PSD2DM		DM2PAS		PSD2PAS		DM2PSD		PAS2PSD		AVG.	
Methods	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF
Single	89.92	82.04	86.47	74.07	90.07	87.53	90.07	87.53	93.76	91.56	93.76	91.56	90.87	75.32	90.87	75.32	90.72	83.12
DFT	90.56	82.86	87.22	75.11	90.51	88.09	90.51	88.09	94.14	92.01	94.14	92.01	91.57	76.49	91.57	76.49	91.28	83.89
TSFT	91.63	83.60	87.98	75.74	91.02	88.66	90.78	88.49	94.51	92.52	94.88	92.83	91.79	76.44	91.89	76.57	91.81	84.36
PE	93.39	**88.41**	88.48	76.13	90.97	88.77	91.04	89.07	94.63	92.64	95.08	93.16	91.62	77.09	92.24	77.63	92.18	85.36
G2GTr	92.56	84.89	87.57	75.45	90.61	88.40	90.87	88.73	94.23	92.18	94.77	92.82	91.94	77.33	91.90	76.35	91.81	84.52
LS	93.52	85.77	88.86	76.82	91.27	88.86	91.19	88.96	94.92	93.04	95.17	93.32	92.51	77.92	92.90	78.02	92.54	85.34
G2GLT	93.23	87.36	88.92	76.92	91.28	88.92	91.63	89.81	94.85	92.90	95.40	93.56	92.42	78.34	92.69	77.58	92.55	85.67
LS+G2GLT	**93.64**	87.47	**89.32**	**77.48**	**91.50**	**89.20**	**91.70**	**89.87**	**95.02**	**93.20**	**95.67**	**93.96**	**93.07**	**79.73**	**92.91**	**78.41**	**92.85**	**86.16**

Table 4: Parsing scores on test data.

ically, G2GTr improves the performance by employing the Graph2Graph Transformer to encode source-annotated information. As the only feature-based method, PE achieves the best average results in all the baselines.

As for our approaches, they achieve comparable conversion scores, and both of them outperform the baselines on average. Besides, they significantly outperform all the baselines in almost all the conversion tasks, except in the conversion from UD-EWT to UD-Enhanced where PE yields the best result. We assume that 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 and they have the highest annotation overlap among all the conversion tasks.[11] While in all the other conversion tasks, the two datasets have completely different label sets, and thus their overlap rate is much lower. Obviously, the feature-based PE approach performs extremely good in the case where the annotation overlap rate is high. However, it is outperformed by our approaches in all the other more complicated conversion tasks where the annotation overlap rate is lower.

Furthermore, since our proposed approaches improve the conversion score in two facets, namely data augmentation and parameter transformation, we assume that the improvements they brought are orthogonal to each other. Therefore, we combine them by simply averaging the arc and label scores they predicted and find that the combined model further significantly improves the performance on all the conversion tasks.

5.2 Utilizing Converted Data

In order to evaluate the quality of the converted data, we utilize them to train a target parser and measure the quality with the parsing score. Specifically, following the data split in Table 2, we first convert **Source-Train** into target-annotated data with different conversion approaches, then train target parsers with the converted data. Eventually, the target parsers are evaluated on the **Test** set.

Table 4 shows the empirical results. Besides the methods in Table 3, we also include a **Single** baseline without the annotation conversion process, which is a target parser trained only on the target annotation in the **Train** set in Table 2. Obviously, utilizing the target data converted from the large-scale source data during training can significantly improve the performance. Moreover, both of our approaches outperform all the baselines in almost all parsing tasks. The result of PE is much higher than that of our approaches for the UD-EWT to UD-Enhanced task. We assume that this is due to the high conversion score of PE on the conversion task from UD-EWT to UD-Enhanced, whose reason has been discussed in Section 5.1. Similar to the case in conversion tasks, the combined model can further improve the parsing performance.

5.3 Effect of Parallel Annotated Data Size

Recall that this paper aims at graph-structured annotation conversion based on an existing source-annotated 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 conversion approaches. 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.[12]

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 almost all the conversion tasks.

[11] Please refer to Appendix A.1 for the annotation overlap information in detail.

[12] We conduct experiments on all conversion tasks except the conversion from UD-EWT to SDG since there are only 1,000 parallel annotated sentences. The results for conversion from PAS to PSD is shown in Appendix A.4.

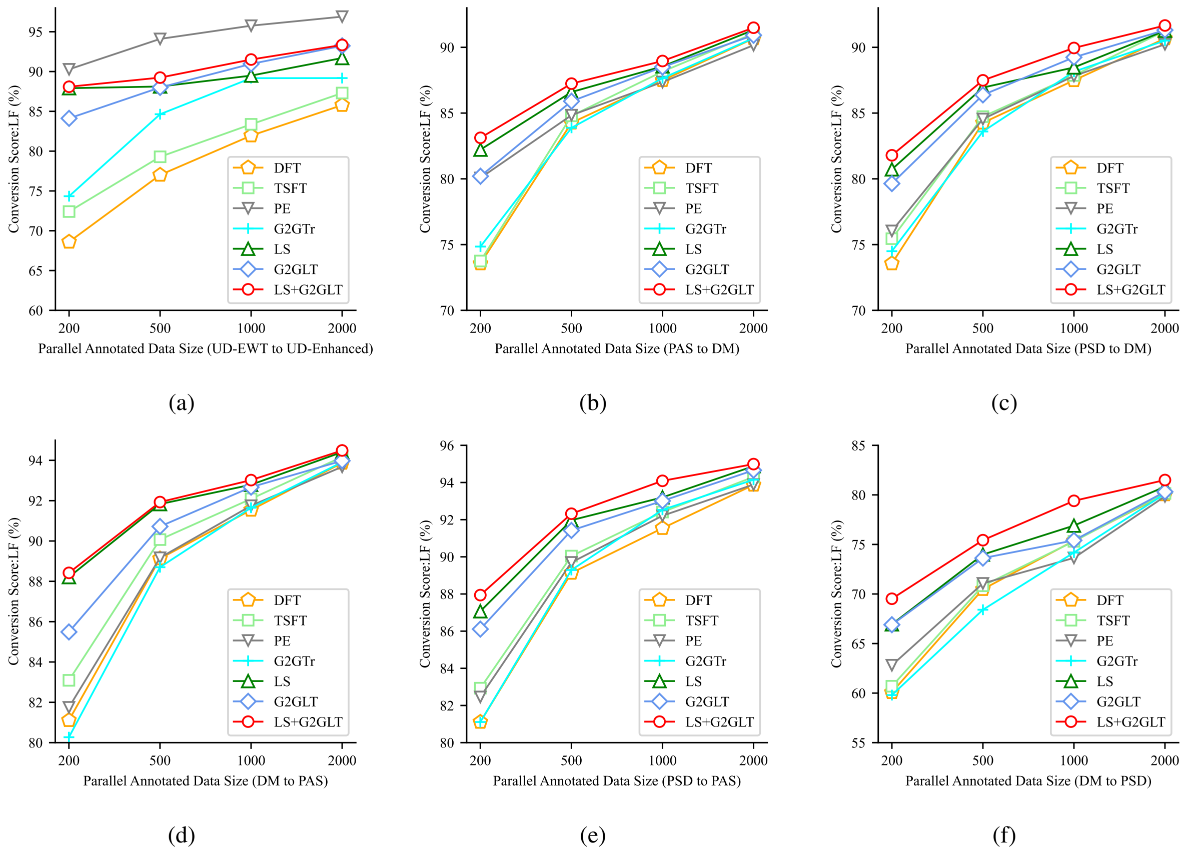

Figure 3: Results for conversions with different parallel annotated data sizes (best viewed in color).

However, the performance of LS, G2GLT 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 high annotation overlap rate between them. With the highly overlapped annotation, our proposed approaches can easily obtain promising results with only 200 parallel annotated sentences. While since the PE method extracts features from the source graph to predict the target graph, it also benefits from the high annotation overlap rate. 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 exactly satisfies the aim of this paper.

6 Conclusion

This paper aims at graph-structured annotation conversion based on an existing source-annotated dataset with little human labour. We propose two graph-to-graph annotation conversion approaches, namely Label Switching and Graph2Graph Linear Transformation, and show their effectiveness on eight annotation conversion tasks and converted data utilizing tasks. Results show that 1) the two approaches achieve comparable conversion scores; 2) our proposed approaches are most suitable for cases where only limited parallel annotated data is available; 3) the two approaches can be combined to further improve the performance.

7 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.

Acknowledgments

This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 62176078.

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 *Proc. of SemEval*.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proc. of ACL*.

Anette Frank. 2001. Treebank conversion - converting the negra treebank to an ltag grammar. In *Proceedings of the Workshop on Multi-layer Corpus-based Analysis*.

Wenbin Jiang, Yajuan Lü, Liang Huang, and Qun Liu. 2015. Automatic adaptation of annotations. *Computational Linguistics*.

Xinzhou Jiang, Zhenghua Li, Bo Zhang, Min Zhang, Sheng Li, and Luo Si. 2018. Supervised treebank conversion: Data and approaches. In *Proc. of ACL*.

Xiang Li, Wenbin Jiang, Yajuan Lü, and Qun Liu. 2013. Iterative transformation of annotation guidelines for constituency parsing. In *Proc. of ACL*.

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.

Alireza Mohammadshahi and James Henderson. 2021. Recursive Non-Autoregressive Graph-to-Graph Transformer for Dependency Parsing with Iterative Refinement. *Transactions of the Association for Computational Linguistics*.

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*.

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*.

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 *Proc. of SemEval*.

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 *Proc. of IJCAI*.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Proc. of LREC*.

Djamé Seddah, Eric de la Clergerie, Benoît Sagot, Héctor Martínez Alonso, and Marie Candito. 2018. Cheating a Parser to Death: Data-driven Cross-Treebank Annotation Transfer. In *Proc. of LREC*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proc. of LREC*.

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 *Proc. of ACL*.

A Appendix

A.1 Annotation Overlap

In this section, we evaluate the annotation overlap between the datasets in our conversion tasks. Specifically, for each of the eight conversion tasks used in this paper, we directly evaluate the original source dataset on the gold target dataset and use the scores to measure the annotation overlap between the two datasets. A higher score between two datasets represents a higher annotation overlap rate between them.

Source	Target	UF	LF
UD-EWT	**UD-Enhanced**	96.95	83.87
UD-EWT	**SDG**	87.98	-
PAS	**DM**	63.78	-
PSD	**DM**	27.21	-
DM	**PAS**	63.78	-
PSD	**PAS**	30.24	-
DM	**PSD**	27.21	-
PAS	**PSD**	30.24	-

Table 5: Annotation overlap in terms of UF and LF.

Results are shown in Table 5, 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. 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 annotation overlap 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 overlap 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 overlap rate between DM, PAS and PSD are much lower, with only 27.21% 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.

A.2 Semantic Dependency Graph Annotation Guidelines

We modified Chinese Semantic Dependency Graph guidelines[13] 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.

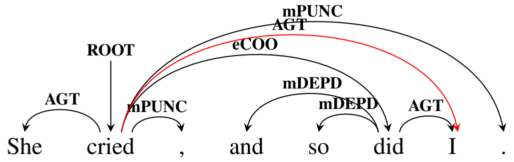

Figure 4: Example of annotation for omitted predicates

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 6: Label set of the semantic relation of EN-SDG

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

	UD2UD-En		UD2SDG		PAS2DM		PSD2DM		DM2PAS		PSD2PAS		DM2PSD		PAS2PSD		AVG.	
Methods	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF	UF	LF
DFT	0.08	0.23	0.06	0.04	0.08	0.10	0.08	0.10	0.10	0.09	0.10	0.09	0.12	0.21	0.12	0.21	0.03	0.08
TSFT	0.13	0.06	0.12	0.17	0.10	0.13	0.15	0.21	0.11	0.16	0.10	0.06	0.1	0.30	0.18	0.32	0.06	0.11
PE	0.08	0.28	0.14	0.17	0.05	0.08	0.10	0.16	0.02	0.07	0.17	0.27	0.15	0.35	0.14	0.25	0.01	0.05
G2GTr	0.24	0.15	0.09	0.20	0.09	0.06	0.08	0.05	0.07	0.05	0.17	0.26	0.09	0.06	0.14	0.18	0.06	0.05
LS	0.35	0.39	0.04	0.05	0.09	0.17	0.16	0.21	0.06	0.10	0.10	0.09	0.03	0.18	0.07	0.05	0.07	0.05
G2GLT	0.13	0.10	0.12	0.17	0.08	0.06	0.07	0.08	0.26	0.32	0.10	0.09	0.28	0.25	0.09	0.36	0.11	0.10
LS+G2GLT	0.16	0.11	0.05	0.13	0.03	0.08	0.07	0.09	0.01	0.01	0.01	0.01	0.08	0.10	0.02	0.22	0.02	0.02

Table 7: Standard Deviation for conversion scores on test data.

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 6.

A.3 Standard Deviation

Table 7 shows the standard deviation for the experiments in Table 3.

A.4 Effect of Parallel Annotated Data Size

In this section, we report the result for conversion from PAS to PSD with different parallel annotated data sizes. The results are shown in Figure 5.

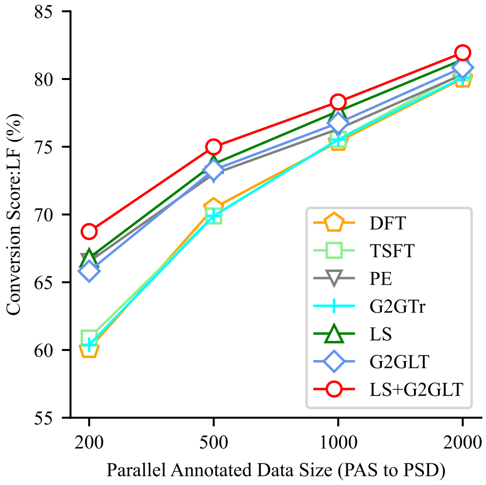

Figure 5: Results for conversions from PAS to PSD with different parallel annotated data sizes.