The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation

(LREC-COLING 2024)

Few-Shot Semantic Dependency Parsing via Graph Contrastive Learning

School of Computer Science and Engineering, Southeast University, China

- Reporter: Bin Li
- Email: lib@seu.edu.cn

目录
CONTENTS

1 Introduction

2 Methodology

3 Experiments

4 Analysis

5 Conclusion

1

Introduction

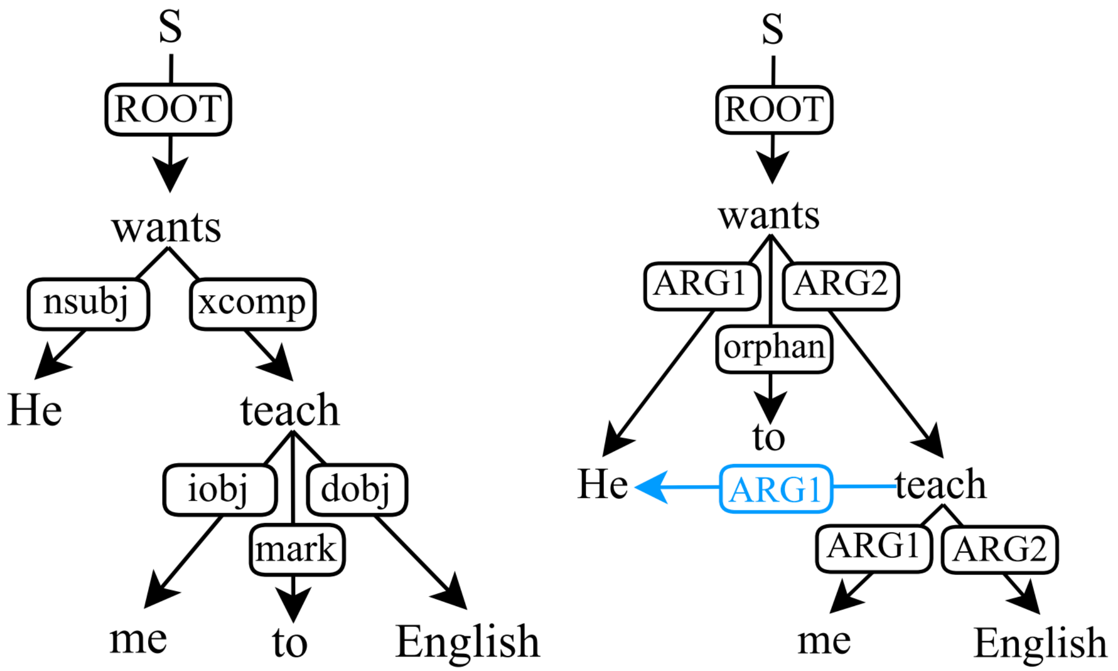
1 Introduction

What is semantic dependency parsing?

Semantic dependency parsing (SDP) is a linguistic task that focuses on capturing intricate bi-lexical relationships, allowing words to have multiple dependency heads, and producing a labeled directed acyclic graph that accurately represents the meaning of the sentence. SDP derives from syntactic dependency parsing which aims to represent the syntactic structure of a sentence through a labeled tree. Hence, there are a lot of similarities between syntactic and semantic dependencies.

Similarities between syntactic and semantic dependencies.

Sentence: He wants to teach me English.



(a) Syntactic Tree (b) Semantic Graph

1 Introduction

Which tasks can semantic dependency parsing be applied?

SDP has been widely applied in many NLP downstream tasks, including:

- ✓ sentiment analysis
- ✓ abstractive summarization
- ✓ dialogue generation
- ✓ natural language understanding
- ✓ ...

- Existing SDP models can be classified as transition-based and graph-based.
- Transition-based models score all transition actions according to the current parsing state and select the highest score transition action in each step. The final semantic dependency graph (SDG) could be incrementally built by a sequence of selected transition actions.
- Graph-based models score each substructure of a potential SDG and utilize exact or approximate decoding algorithms to search the highest-scoring SDG. Among them, graph neural networks (GNNs) based models are especially successful because of their powerful graph representation learning ability.

1 Introduction

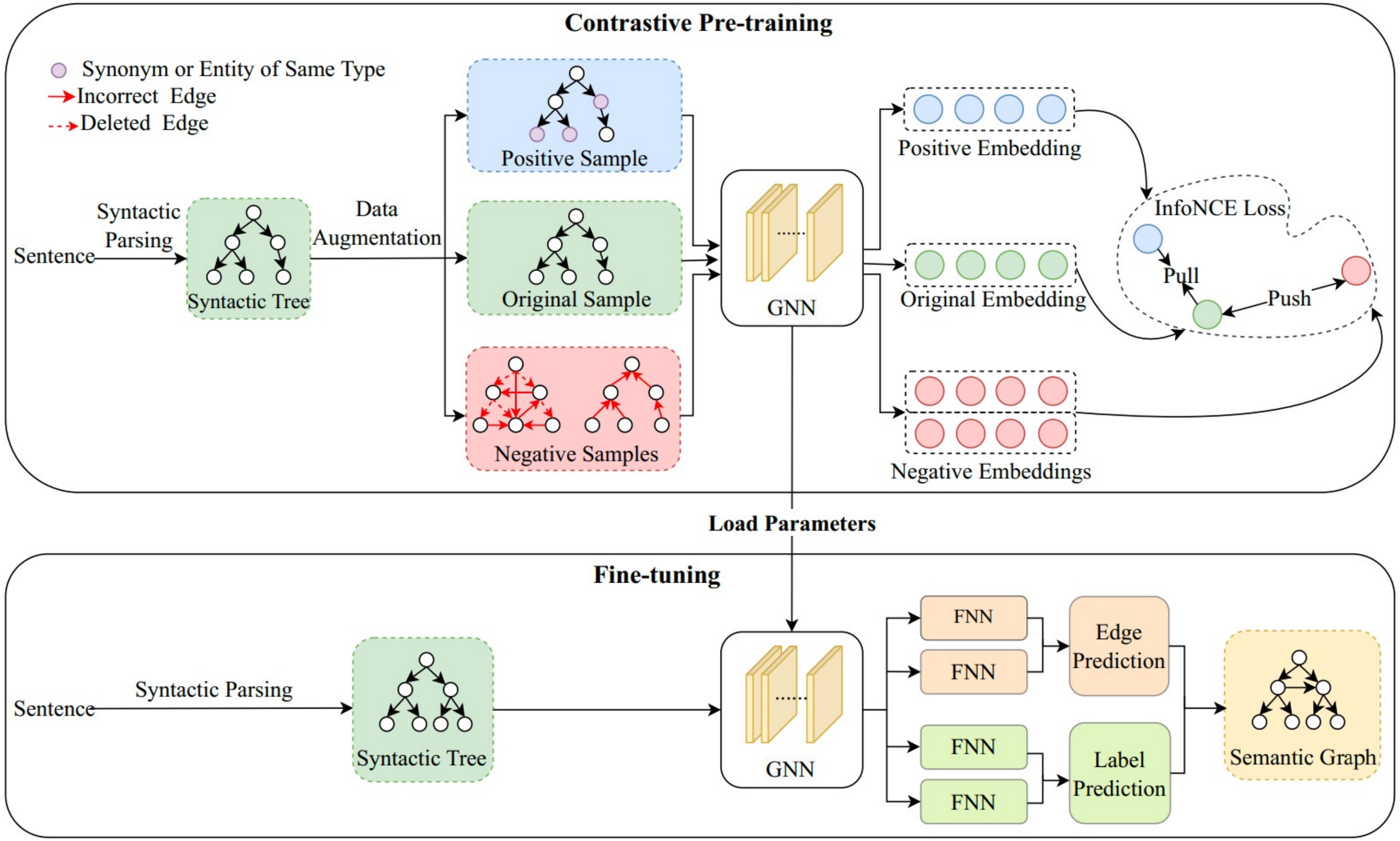
- Although the benefits provided by SDP and the remarkable performance achieved by previous studies, training a high-performing SDP model requires large amounts of labeled data. This issue becomes more severe with the rise of GNNs because GNN-based models are more data-hungry and susceptible to over-fitting when lacking training data. To alleviate this drawback, a semi-supervised model is presented. This model leverages both labeled and unlabeled data to learn a dependency graph parser.
- Another study leverages a multitask learning framework coupled with annotation projection for languages without SDP annotated. They use annotation projection to transfer semantic annotations from a source language to the target language. These two attempts alleviate the data-hungry issue to some extent, but their performances are still not satisfactory.
- Recently, contrastive learning, a category of self-supervised learning (SSL), has emerged as a new paradigm for making use of large amounts of unlabeled data when labeled data is limited. Contrastive learning aims to learn the representation by concentrating positive pairs and pushing negative pairs apart.
- Motivated by plenty of similarities between syntactic and semantic dependencies and the success of contrast learning in few-shot learning, we propose a syntactic dependency-guided graph contrastive learning framework for few-shot SDP (SynGCL-SDP) in this paper.

2

Methodology

2 Methodology

- There are two stages to perform few-shot SDP: the unsupervised pre-training and the supervised fine-tuning.



- Contrastive Samples Construction
  - Original Graph Construction
  - Positive Graph Construction
  - Negative Graph Construction
- Contrastive Pre-training
- Fine-tuning

2 Methodology

Contrastive Samples Construction

- Original Graph Construction

we collect plenty of unlabeled sentences from machine translation corpus and adopt a well-trained parsing model - Stanza 1 to automatically generate a syntactic dependency tree as the original graph for each sentence.

- Positive Graph Construction

(1) replacing the nodes (tokens) of the original graph with synonymous words.

(2) replacing the nodes corresponding to the recognized named entity in the sentence with the named entity of the same type in the dictionary.

- Negative Graph Construction

(1) deleting all correct dependency edges and randomly adding incorrect dependency edges for node pairs that have no dependency relations.

(2) exchanging the head and dependency node of each dependency edge.

(3) changing the dependency node to another randomly chosen node for each dependency edge.

2 Methodology

Contrastive Pre-training

- **Contextualized Representation Learning**

$$x_i = e_i^{(word)} \oplus e_i^{(tag)} \oplus e_i^{(lemma)} \oplus e_i^{(char)}$$

$$c_i = BiLSTM(x_i)$$

- **Graph Representation Learning**

$$H^{(k)} = \text{GNNLayer}^{(k-1)}(H^{(k-1)}, A) \qquad H = (h_1, h_2, \cdots, h_n)$$

- **Contrastive Embedding**

$$z_i = c_i \oplus h_i$$

- **Pre-training Objective**

$$\mathcal{L}_{pt} = -\log \frac{exp(sim(z, z^+))/\tau}{\sum_{z^- \in Z^-} exp(sim(z, z^-))/\tau}$$

2 Methodology

Fine-tunning

- **Hidden State**

$$h_i^{(edge-head)} = \mathit{FNN}^{(edge-head)}(z_i)$$

$$h_i^{(label-head)} = \mathit{FNN}^{(label-head)}(z_i)$$

$$h_i^{(edge-dep)} = \mathit{FNN}^{(edge-dep)}(z_i)$$

$$h_i^{(label-dep)} = \mathit{FNN}^{(label-dep)}(z_i)$$

- **Estimated Value**

$$s_{i,j}^{(edge)} = Biaff^{(edge)}(h_i^{(edge-dep)}, h_j^{(edge-head)})$$

$$\hat{y}_{i,j}^{(edge)} = \{s_{i,j}^{(edge)} > 0\}$$

$$s_{i,j}^{(label)} = Biaff^{(label)}(h_i^{(label-dep)}, h_j^{(label-head)})$$

$$\hat{y}_{i,j}^{(label)} = \arg\max s_{i,j}^{(label)}$$

- **Fine-tuning Objective**

$$\mathcal{L}^{(edge)}(\theta_1) = CE(\hat{y}_{i,j}^{(edge)}, y_{i,j}^{(edge)})$$

$$\mathcal{L}^{(label)}(\theta_2) = CE(\hat{y}_{i,j}^{(label)}, y_{i,j}^{(label)})$$

$$\mathcal{L}_{ft} = \alpha\mathcal{L}^{(edge)} + (1-\alpha)\mathcal{L}^{(label)}$$

3

Experiments

3 Experiments

Dataset

SemEval-2015 Task 18 dataset

- English
- 3 different formalisms:
  - DELPH-IN MRS (DM)
  - Predicate-Argument Structure (PAS)
  - Prague Semantic Dependencies (PSD)

Evaluation Metrics

UnLabeled F-measure score (UF1) and **Labeled F-measure score** (LF1) are used as the metrics to evaluate our parser's performance.

3 Experiments

Unlabeled Data for Contrastive Pre-training

Unlabeled raw sentences are downloaded from the WMT14 machine translation monolingual training data. (http://statmt.org/wmt14/training-monolingual-news-crawl/news.2010.en.shuffled.gz)

Few-Shot Data Sampling

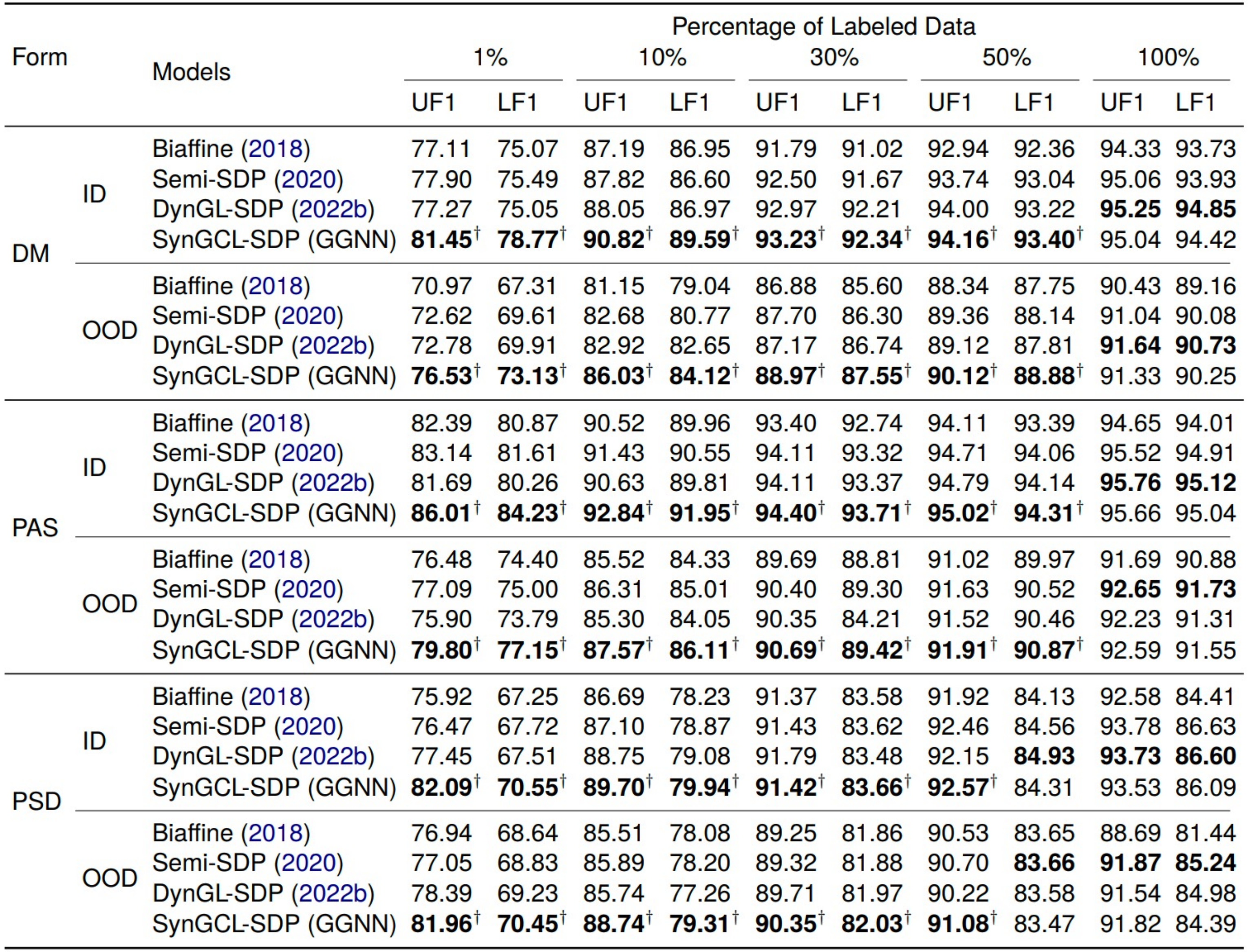
5 sampling rates:

1%   10%   30%   50%   100%

Compared Approaches

1. Biaffine (Dozat and Manning, 2018) is a simple but accurate supervised model.
2. Semi-SDP (Jia et al., 2020) is a semi-supervised model which aims to improve performance with both the labeled and unlabeled data.
3. DynGL-SDP (Li et al., 2022b) is a dynamic graph learning-based model, which also achieves the start-of-the-art (SOTA) performance.

3 Experiments



Form		Models	Percentage of Labeled Data									
			1%		10%		30%		50%		100%	
			UF1	LF1	UF1	LF1	UF1	LF1	UF1	LF1	UF1	LF1
DM	ID	Biaffine (2018)	77.11	75.07	87.19	86.95	91.79	91.02	92.94	92.36	94.33	93.73
		Semi-SDP (2020)	77.90	75.49	87.82	86.60	92.50	91.67	93.74	93.04	95.06	93.93
		DynGL-SDP (2022b)	77.27	75.05	88.05	86.97	92.97	92.21	94.00	93.22	**95.25**	**94.85**
		SynGCL-SDP (GGNN)	**81.45**†	**78.77**†	**90.82**†	**89.59**†	**93.23**†	**92.34**†	**94.16**†	**93.40**†	95.04	94.42
	OOD	Biaffine (2018)	70.97	67.31	81.15	79.04	86.88	85.60	88.34	87.75	90.43	89.16
		Semi-SDP (2020)	72.62	69.61	82.68	80.77	87.70	86.30	89.36	88.14	91.04	90.08
		DynGL-SDP (2022b)	72.78	69.91	82.92	82.65	87.17	86.74	89.12	87.81	**91.64**	**90.73**
		SynGCL-SDP (GGNN)	**76.53**†	**73.13**†	**86.03**†	**84.12**†	**88.97**†	**87.55**†	**90.12**†	**88.88**†	91.33	90.25
PAS	ID	Biaffine (2018)	82.39	80.87	90.52	89.96	93.40	92.74	94.11	93.39	94.65	94.01
		Semi-SDP (2020)	83.14	81.61	91.43	90.55	94.11	93.32	94.71	94.06	95.52	94.91
		DynGL-SDP (2022b)	81.69	80.26	90.63	89.81	94.11	93.37	94.79	94.14	**95.76**	**95.12**
		SynGCL-SDP (GGNN)	**86.01**†	**84.23**†	**92.84**†	**91.95**†	**94.40**†	**93.71**†	**95.02**†	**94.31**†	95.66	95.04
	OOD	Biaffine (2018)	76.48	74.40	85.52	84.33	89.69	88.81	91.02	89.97	91.69	90.88
		Semi-SDP (2020)	77.09	75.00	86.31	85.01	90.40	89.30	91.63	90.52	**92.65**	**91.73**
		DynGL-SDP (2022b)	75.90	73.79	85.30	84.05	90.35	84.21	91.52	90.46	92.23	91.31
		SynGCL-SDP (GGNN)	**79.80**†	**77.15**†	**87.57**†	**86.11**†	**90.69**†	**89.42**†	**91.91**†	**90.87**†	92.59	91.55
PSD	ID	Biaffine (2018)	75.92	67.25	86.69	78.23	91.37	83.58	91.92	84.13	92.58	84.41
		Semi-SDP (2020)	76.47	67.72	87.10	78.87	91.43	83.62	92.46	84.56	93.78	86.63
		DynGL-SDP (2022b)	77.45	67.51	88.75	79.08	91.79	83.48	92.15	**84.93**	**93.73**	**86.60**
		SynGCL-SDP (GGNN)	**82.09**†	**70.55**†	**89.70**†	**79.94**†	**91.42**†	**83.66**†	**92.57**†	84.31	93.53	86.09
	OOD	Biaffine (2018)	76.94	68.64	85.51	78.08	89.25	81.86	90.53	83.65	88.69	81.44
		Semi-SDP (2020)	77.05	68.83	85.89	78.20	89.32	81.88	90.70	**83.66**	**91.87**	**85.24**
		DynGL-SDP (2022b)	78.39	69.23	85.74	77.26	89.71	81.97	90.22	83.58	91.54	84.98
		SynGCL-SDP (GGNN)	**81.96**†	**70.45**†	**88.74**†	**79.31**†	**90.35**†	**82.03**†	**91.08**†	83.47	91.82	84.39

Main Results

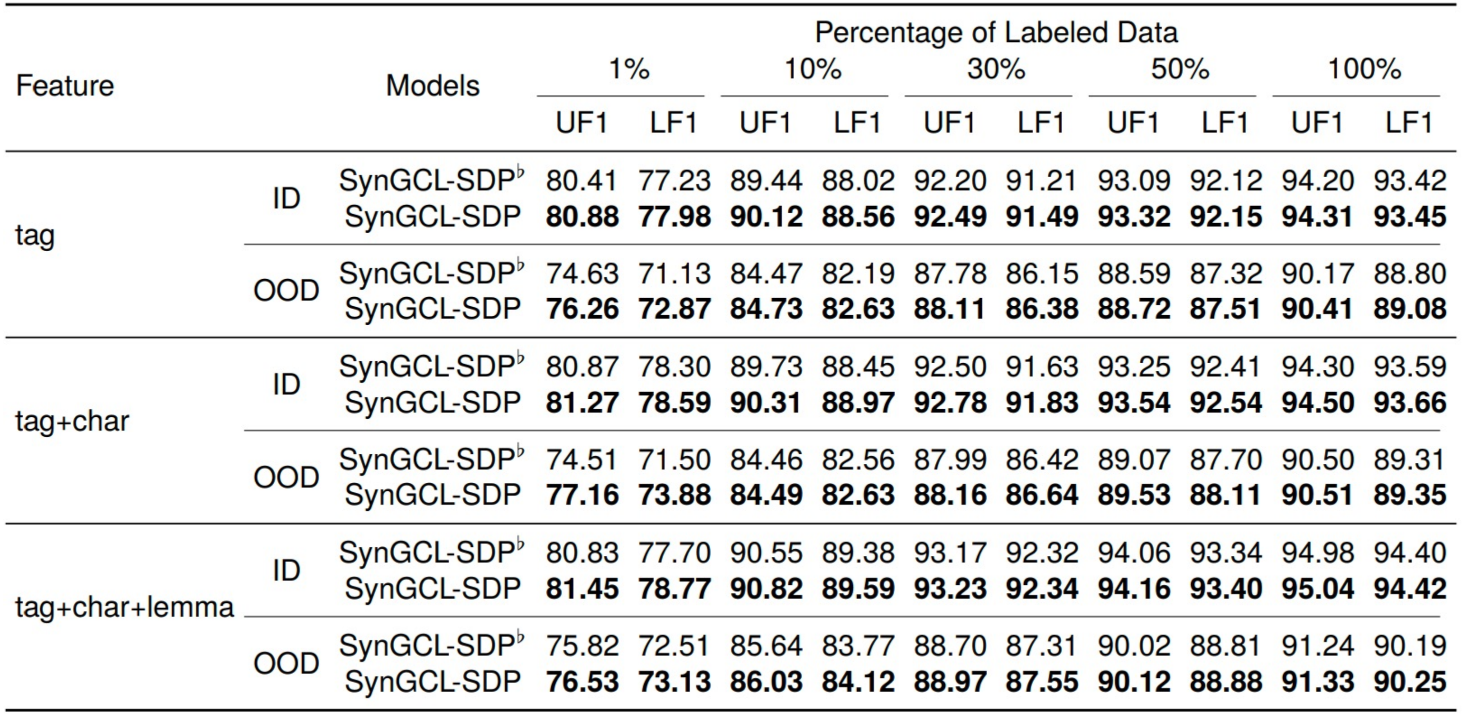
- From the main results, we can see that our proposed model performs better than the compared models on most few-shot data groups, especially with the 1% labeled data (only 339 labeled sentences are used). This highly suggests that our model is superior in few-shot SDP.
- Particularly, benefiting from the pretraining stage on plenty of the unlabeled sentences, our model shows more advantages on the OOD test sets, which suggests the good generalization of our model

4

Analysis

4 Analysis

Effect of Pre-training and Features

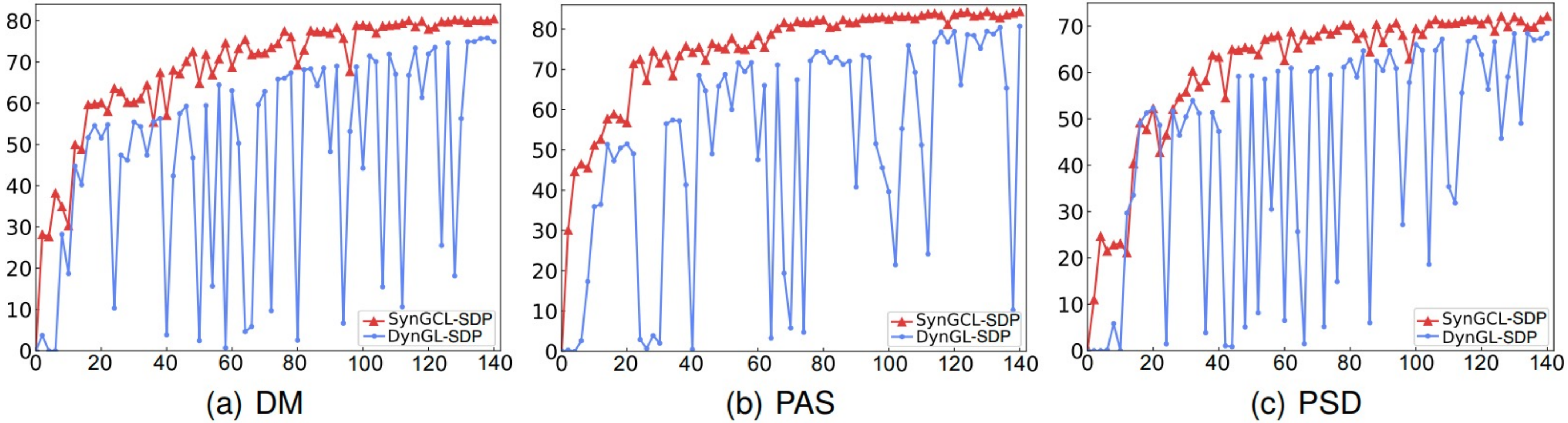


Feature		Models	Percentage of Labeled Data									
			1%		10%		30%		50%		100%	
			UF1	LF1	UF1	LF1	UF1	LF1	UF1	LF1	UF1	LF1
tag	ID	SynGCL-SDP$^\flat$	80.41	77.23	89.44	88.02	92.20	91.21	93.09	92.12	94.20	93.42
		SynGCL-SDP	**80.88**	**77.98**	**90.12**	**88.56**	**92.49**	**91.49**	**93.32**	**92.15**	**94.31**	**93.45**
	OOD	SynGCL-SDP$^\flat$	74.63	71.13	84.47	82.19	87.78	86.15	88.59	87.32	90.17	88.80
		SynGCL-SDP	**76.26**	**72.87**	**84.73**	**82.63**	**88.11**	**86.38**	**88.72**	**87.51**	**90.41**	**89.08**
tag+char	ID	SynGCL-SDP$^\flat$	80.87	78.30	89.73	88.45	92.50	91.63	93.25	92.41	94.30	93.59
		SynGCL-SDP	**81.27**	**78.59**	**90.31**	**88.97**	**92.78**	**91.83**	**93.54**	**92.54**	**94.50**	**93.66**
	OOD	SynGCL-SDP$^\flat$	74.51	71.50	84.46	82.56	87.99	86.42	89.07	87.70	90.50	89.31
		SynGCL-SDP	**77.16**	**73.88**	**84.49**	**82.63**	**88.16**	**86.64**	**89.53**	**88.11**	**90.51**	**89.35**
tag+char+lemma	ID	SynGCL-SDP$^\flat$	80.83	77.70	90.55	89.38	93.17	92.32	94.06	93.34	94.98	94.40
		SynGCL-SDP	**81.45**	**78.77**	**90.82**	**89.59**	**93.23**	**92.34**	**94.16**	**93.40**	**95.04**	**94.42**
	OOD	SynGCL-SDP$^\flat$	75.82	72.51	85.64	83.77	88.70	87.31	90.02	88.81	91.24	90.19
		SynGCL-SDP	**76.53**	**73.13**	**86.03**	**84.12**	**88.97**	**87.55**	**90.12**	**88.88**	**91.33**	**90.25**

- To investigate the effect of pre-training stage and the effect of each type of feature embedding, we conduct a controlled experiment on the SemEval- 2015 Task 18 English dataset in DM formalism with the combination of three types of features, in which the one loads the pre-trained GNNs for initialization (SynGCL-SDP) and another not (SynGCL-SDP^b).
- From the result. We can see that the performance of the model that uses the pre-trained GNNs for initialization outperforms the model that doesn’ t use the pre-trained GNNs in all few-shot sampling groups and all combinations of the three types of features. Moreover, with the increase of the labeled SDP data, the advantage of SynGCL-SDP gradually decreases.

4 Analysis

Convergence Behavior



- The above Figure compares the convergence curves of our model (SynGCL-SDP) and the GNN-based SOTA model (DynGL-SDP) when using 1% training data.
- From the compared curves, we can clearly see that the performance of DynGL-SDP is quite unstable during the training process when there are very few labeled samples available, meaning that it is susceptible to overfitting.
- On the contrary, the performance of our model improves steadily until it converges as the number of training epochs increases, indicating that our model is still stable and not prone to over-fitting when few labeled data . available.

5

Conclusion

5 Conclusion

- In this paper, we propose a syntax-guided graph contrastive learning framework for few-shot SDP.
- The proposed framework pre-trains GNNs with plenty of unlabeled data and fine-tunes the pretrained GNNs with few-shot labeled SDP data.
- The pretrained GNNs can also take advantage of large amounts of unlabeled data to adapt to out-of-domain.
- Extensive evaluations on SemEval-2015 Task 18 English dataset in three formalisms show that our model performs better when limited data is available.

THANK YOU!