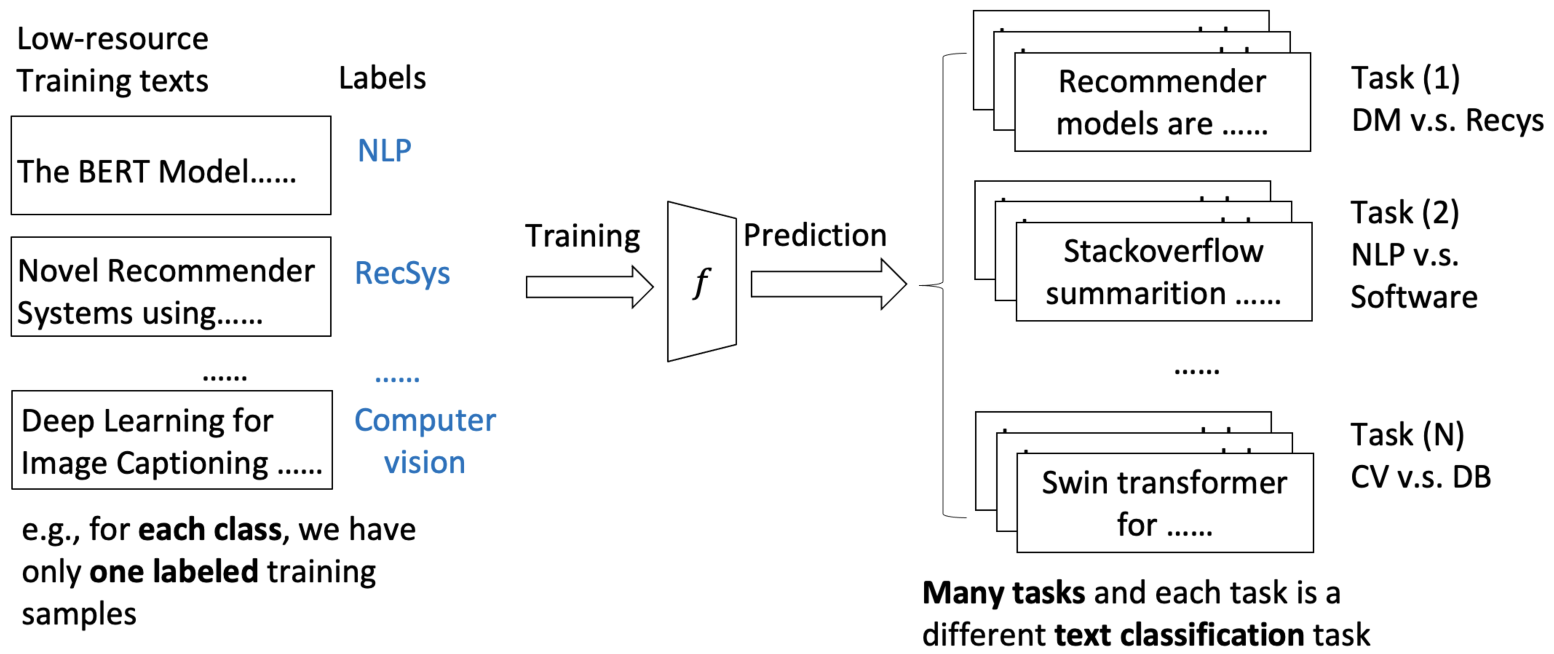
Augmenting Low-Resource Text Classification with Graph-Grounded Pre-training and Prompting

Zhihao Wen and Yuan Fang Singapore Management University

SMU School of Computing and Information Systems

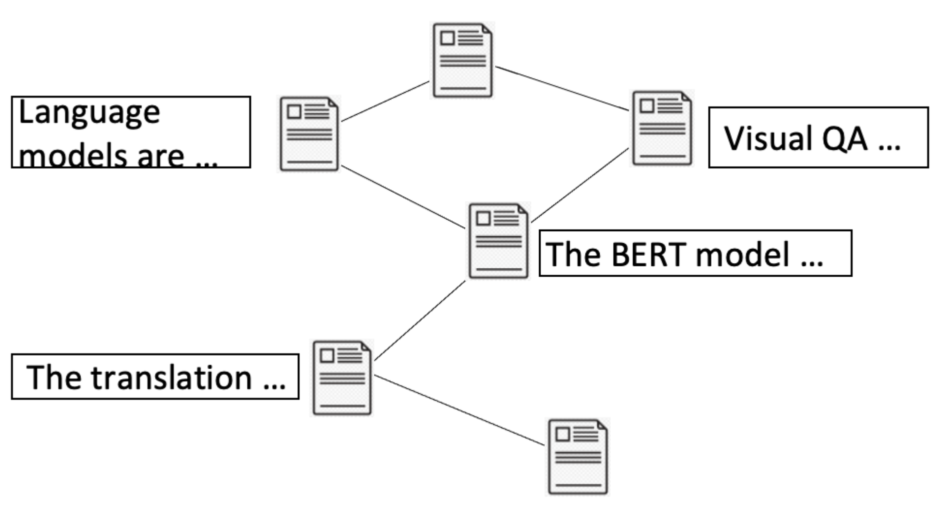
Introduction



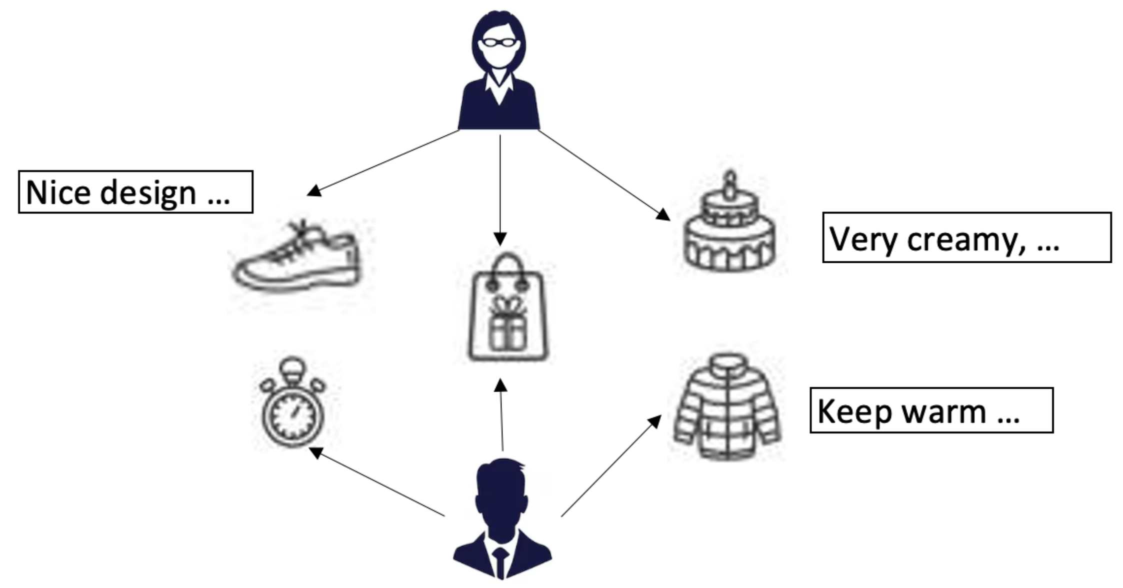
- Low-resource text classification, in which **no or few labeled** samples are available, is an important research problem.
- When there are lots of downstream tasks, developing **parameter and time efficient tuning method** holds significant practical implications.

Motivation

- Text data are frequently grounded on **network structures**, exposing valuable relationships, which can be used to augment low-resource text classification.
- While existing pre-trained language models and prompting do not exploit these relationships, graph neural networks (**GNNs**) are designed to learn from graph structures based on a message-passing architecture.

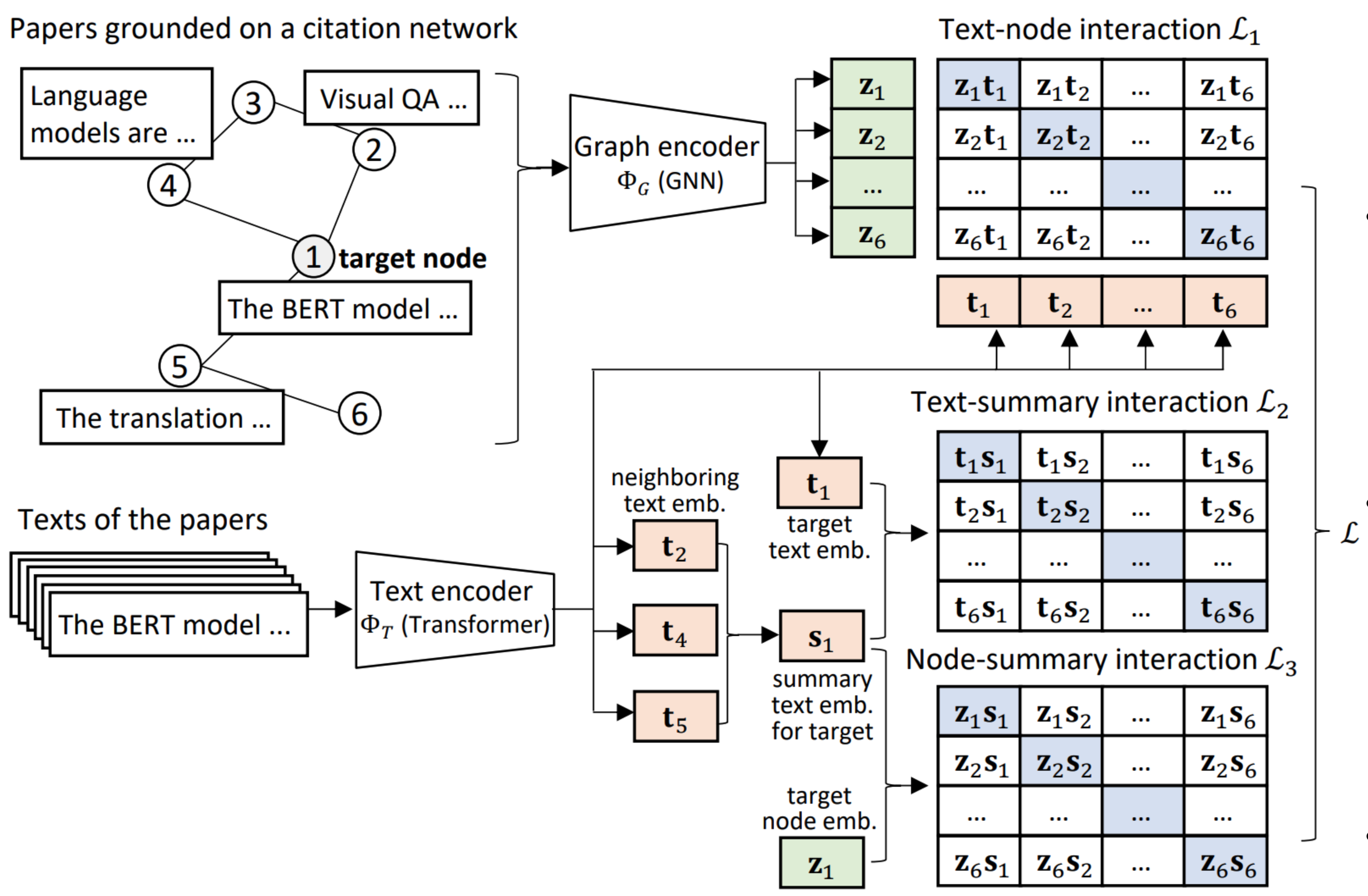


Citation graph for online articles



E-commerce item review graph

Graph grounded contrastive pre-training

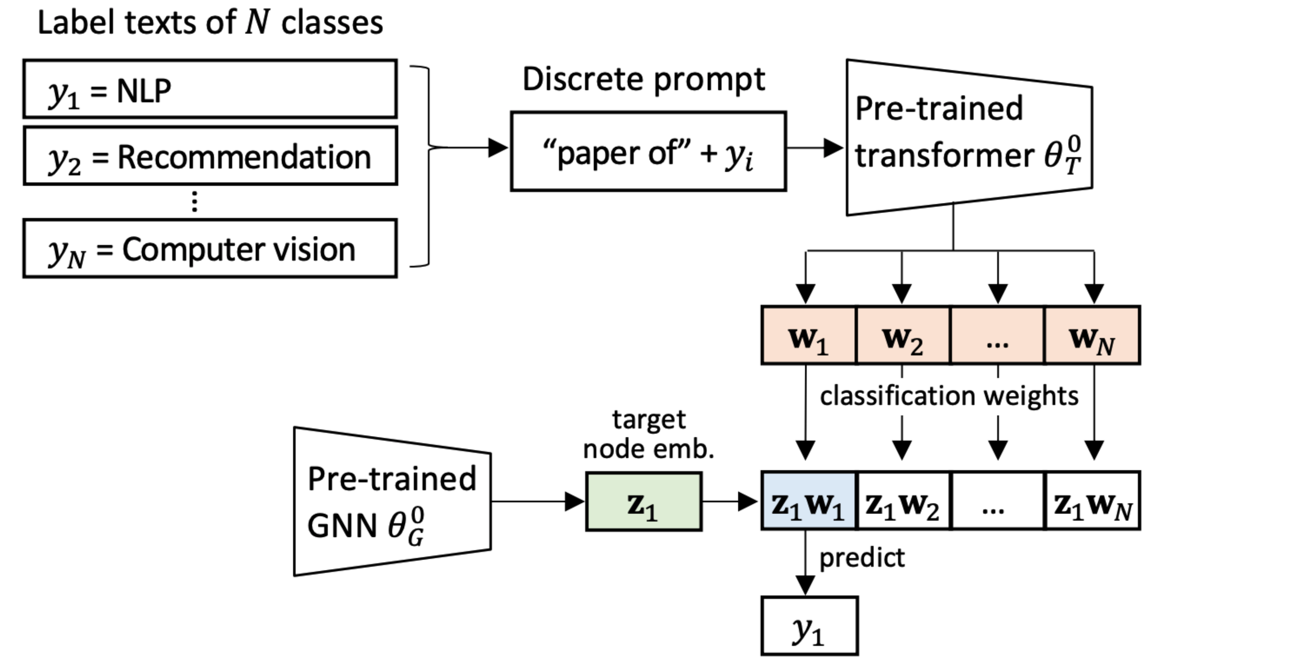


(a) Graph-grounded contrastive pre-training

- During pre-training, we learn a **dual-modal** embedding space by jointly training a text encoder and graph encoder in a self-supervised fashion through **3 contrastive strategies**.
- Strategy 1: text-node interaction. Predict the text of a document matches which node in the graph. We maximize the cosine similarity of n matching pairs, while minimizing that of the n^2 − n unmatching pairs.
- Strategy 2: text-summary interaction. Each document has a set of neighboring documents defined by graph topology. The neighboring documents are a summary of the target document. We align the text embedding and its corresponding summary text embedding.
- Strategy 3: node-summary interaction. Align the **node** embedding $\mathbf{z}_i$ and its neighborhood-based **summary** text embedding $\mathbf{s}_i$.

Prompt-assisted text classification

Discrete prompt for zero-shot classification



- Predict the class whose **label text** embedding has the **highest similarity** to the **node** embedding
- **Classification weights** can be generated by the **text encoder** based on the **class label texts**

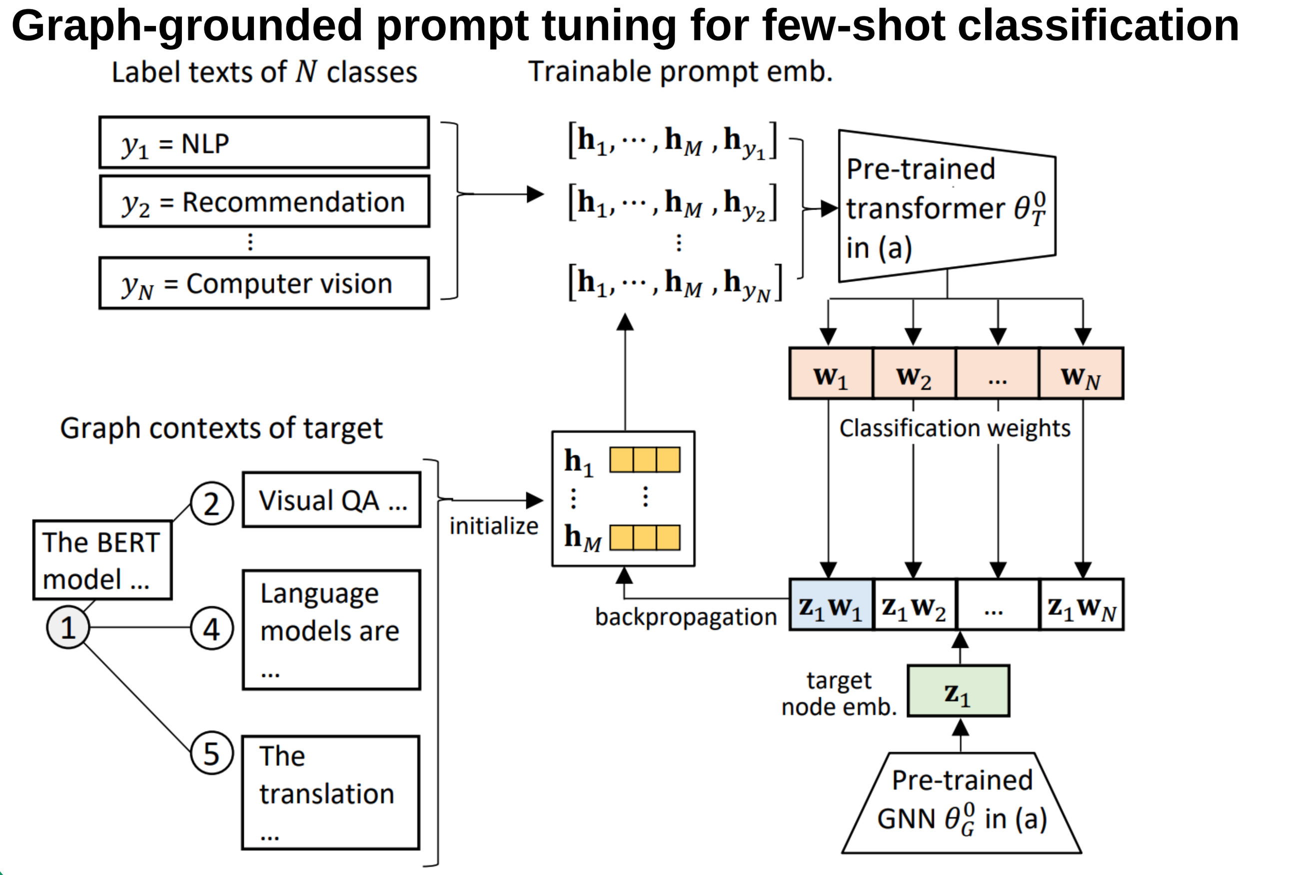
$$\mathbf{w}_y = \phi_T(\text{"prompt [CLASS]"}; \theta_T^0)$$

e.g., "A paper of " label text, e.g., "NLP"

- Class distribution is predicted as

$$p(y \mid \mathbf{z}_i) = \frac{\exp(\langle \mathbf{z}_i, \mathbf{w}_y \rangle)}{\sum_{y=1}^{N} \exp(\langle \mathbf{z}_i, \mathbf{w}_y \rangle)}$$

Graph-grounded prompt tuning for few-shot classification



- Discrete prompts are difficult to optimize.
- Resort to **prompt tuning**, substituting discrete prompts with **learnable continuous** vectors, while keeping the parameters of PLM **frozen**
- Instead of a sequence of **discrete tokens**, we use a sequence of **continuous embeddings**

$$\mathbf{w}_y = \phi_T([\mathbf{h}_1, \cdots, \mathbf{h}_M, \mathbf{h}_{\text{CLASS}}]; \theta_T^0)$$

- We initialize the prompt embeddings with **graph contexts**.
- A node v_i and its neighbor set $\{v_j | j \in \mathcal{N}_i\}$ are collectively called the *graph contexts* of v_i .

Experiment & Conclusion

Five-shot **classification performance (percent) with 95% confidence intervals.**

		Cora		Art		Industrial		M.I.	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
End-to-end GNN	GCN	41.15±2.41	34.50±2.23	22.47±1.78	15.45±1.14	21.08±0.45	15.23±0.29	22.54±0.82	16.26±0.72
	SAGE_{sup}	41.42±2.90	35.14±2.14	22.60±0.56	16.01±0.28	20.74±0.91	15.31±0.37	22.14±0.80	16.69±0.62
	TextGCN	59.78±1.88	55.85±1.50	43.47±1.02	32.20±1.30	53.60±0.70	45.97±0.49	46.26±0.91	38.75±0.78
Pre-trained GNN	GPT-GNN	76.72±2.02	72.23±1.17	65.15±1.37	52.79±0.83	62.13±0.65	54.47±0.67	67.97±2.49	59.89±2.51
	DGI	<u>78.42</u>±1.39	<u>74.58</u>±1.24	65.41±0.86	53.57±0.75	52.29±0.66	45.26±0.51	68.06±0.73	60.64±0.61
	$\text{SAGE}_{\text{self}}$	77.59±1.71	73.47±1.53	76.13±0.94	65.25±0.31	71.87±0.61	65.09±0.47	<u>77.70</u>±0.48	<u>70.87</u>±0.59
Pre-trainned Transformers	BERT	37.86±5.31	32.78±5.01	46.39±1.05	37.07± 0.68	54.00±0.20	47.57±0.50	50.14±0.68	42.96±1.02
	BERT*	27.22±1.22	23.34±1.11	45.31±0.96	36.28±0.71	49.60±0.27	43.36±0.27	40.19±0.74	33.69±0.72
	RoBERTa	62.10±2.77	57.21±2.51	72.95±1.75	62.25±1.33	76.35±0.65	70.49±0.59	70.67±0.87	63.50±1.11
	RoBERTa*	67.42±4.35	62.72±3.02	74.47±1.00	63.35±1.09	77.08±1.02	71.44±0.87	74.61±1.08	67.78±0.95
Prompt tuning	P-Tuning v2	71.00±2.03	66.76±1.95	<u>76.86</u>±0.59	<u>66.89</u>±1.14	<u>79.65</u>±0.38	<u>74.33</u>±0.37	72.08±0.51	65.44±0.63
	G2P2-p	79.16±1.23	74.99±1.35	79.59±0.31	68.26±0.43	80.86±0.40	74.44±0.29	81.26±0.36	74.82±0.45
	G2P2	**80.08***±1.33	**75.91***±1.39	**81.03***±0.43	**69.86***±0.67	**82.46***±0.29	**76.36***±0.25	**82.77***±0.32	**76.48***±0.52
	(improv.)	(+2.12%)	(+1.78%)	(+5.43%)	(+4.44%)	(+3.53%)	(+2.7%)	(+6.53%)	(+7.92%)

- G2P2 outperforms the best baseline by around 3–7%.
- Pre-trained/self-supervised models tend to perform better than the end-to-end models,
- PLMs are generally superior to GNNs
- Continuous prompt approach has the advantage of being much cheaper than fine-tuning.
- G2P2-p without prompt tuning is inferior to G2P2

Zero-shot **classification performance**

	Cora	Art	Industrial	M.I.
RoBERTa	30.46±2.01	42.80±0.94	42.89±0.97	36.40±1.20
RoBERTa*	39.58±1.26	34.77±0.65	37.78±0.32	32.17±0.68
RoBERTa*+d	<u>45.53</u>±1.33	36.11±0.66	39.40±1.22	37.65±0.33
BERT	23.58±1.88	35.88±1.44	37.32±0.85	37.42±0.80
BERT*	23.38±1.96	54.27±1.85	<u>56.02</u>±1.22	50.19±0.72
BERT*+d	26.65±1.71	<u>56.61</u>±1.76	55.93±0.96	<u>52.13</u>±0.88
G2P2	63.52±2.89	76.52±0.59	76.66±0.31	74.60±0.62
G2P2+d	**65.28***±3.12	**76.99***±0.60	**77.43***±0.27	**75.86***±0.69
(improv.)	(+45.38%)	(+36.00%)	(+38.22%)	(+45.52%)

- G2P2 and G2P2+d significantly outperforms the baselines.
- Handcrafted discrete **prompts** (i.e., BERT*+d and G2P2+d) can be superior to no prompt (BERT* and G2P2).

Conclusion

- Addressed the problem of **low-resource multi-task text classification**;
- Proposed G2P2, consisting of **three graph interaction-based** contrastive strategies in pre-training, and a **prompting** mechanism for the jointly pre-trained graph-text model in downstream classification.