LEARNING TO TRANSFER VIA MODELLING MULTI-LEVEL TASK DEPENDENCY

Anonymous authors
Paper under double-blind review

1 ABSTRACT

Multi-task learning has been successful in modeling multiple related tasks with large, carefully curated labeled datasets. By leveraging the relationships among different tasks, multi-task learning framework can improve the performance significantly. However, most of the existing works are under the assumption that the predefined tasks are related to each other. Thus, their applications on real-world are limited, because rare real-world problems are closely related. Besides, the understanding of relationships among tasks has been ignored by most of the current methods. Along this line, we propose a novel multi-task learning framework - Learning To Transfer Via Modelling Multi-level Task Dependency, which constructed attention based dependency relationships among different tasks. At the same time, the dependency relationship can be used to guide what knowledge should be transferred, thus the performance of our model also be improved. To show the effectiveness of our model and the importance of considering multi-level dependency relationship, we conduct experiments on several public datasets, on which we obtain significant improvements over current methods.

2 INTRODUCTION

Multitask learning (Caruana, 1997) aims to train a single model on multiple related tasks jointly, so that useful knowledge learned from one task can be transferred to enhance the generalization performances of other tasks. Over last few years, different types of multitask learning mechanism (Sener & Koltun, 2018; Guo & Farooq, 2018; Ish, 2016; Lon, 2015) has been proposed,and have been proved better than single task learning method from natural language processing (Palmer et al., 2017) and computer vision (Cortes et al., 2015) to chemical study (Ramsundar et al., 2015).

Despite the success of multitask learning, most of the current multitask learning framework rely on the assumption that all the tasks are highly correlated and should contribute equally to the model. However, this assumption is not always true in many real-world problems. As a result, the shared parameters might receive contradicting optimization guidance from irrelevant tasks, and thus cannot learn quite well, not even better than single-task models (Standley et al., 2019).

Intuitively, this problem can be alleviated by modelling the hidden dependency structure between tasks. A reasonable multitask learning framework should learn to capture the task dependency automatically, and only transfer the knowledge from relevant source tasks to the target tasks. Moreover, task dependency can be different for different data samples. For example, when we want to predict chemical properties of a particular toxic molecule, its representation learned from toxicity prediction task should be significant to all the other tasks, and thus should be more emphasized.

In this paper, to accurately model the task dependency in both general level and data-specific level, we propose a novel framework, 'Learning to Transfer via ModellIng mulTi-level Task dEpeNdency' (L2T-MITTEN). We firstly model the general task dependency as a weighted graph, as shown in Figure 1, where the dependency of a source task to a target task is implied by the corresponding edge weight. Afterwards, we model the data-specific task dependency, by leveraging multi-task attention mechanism. For a particular data sample, we first get its representations from multiple task-specific bottom models, then calculate the inter-task attention matrix with task-specific projections. Finally, we combine both the general and data-specific task dependency matrix to ensemble multi-task representations. As all these components are fully differentiable, the framework can learn to transfer among multiple tasks end-to-end.

We validate our multi-task learning framework extensively on different tasks, including graph classication, node classification and text classification. Our method outperform all the other state-of-the-art (SOTA) multitask methods. Besides, we show that L2T-MITTEN can be used as an analytic tool to extract interpretable task dependency structure on real-world datasets.

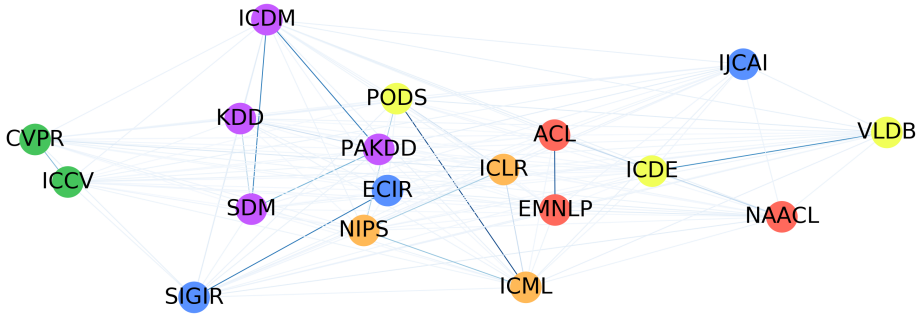


Figure 1: General Task Dependency Relationship

3 Related Work

According to a recent survey (Ruder, 2017), existing multi-task learning methods can be categorized by whether they share the parameters hardly or softly.

3.1 Hard Parameter Sharing

For hard parameter sharing, a bottom network will be shared among all the tasks, and each individual task has their own task-specific output network. The parameter sharing of the bottom network reduce the parameter needed to be learned, and thus can avoid overfitting to a specific task. However, when the tasks are not relevant enough (Sha, 2002; Baxter, 2011), the shared-bottom layers will suffer from optimization conflicts caused by mutually contradicted tasks. If the bottom model is not capable enough to encode all the necessary knowledge from different tasks, this method will fail to correctly capture any of the tasks.

Besides, Dy & Krause (2018) points out that the gradients of some dominant task will be relatively larger than gradients of all the other tasks. Such dominance phenomenon is obvious when the label ratio is imbalanced, with which the model are majorly optimized on data-rich tasks. To alleviate this problem, some recent works (Sener & Koltun, 2018; Dy & Krause, 2018) try to dynamically adjust the task weight during training stage. Sener & Koltun (2018) cast the multitask learning to multi-objective optimization problem, and they use gradient based optimization method to find a Pareto optimal solution. Dy & Krause (2018) proposes a new normalization method on gradients, which attempts to balance the influences of different tasks. Recently, Guo & Farooq (2018) proposes to apply Mixture-of-Experts on multitask learning, which linearly combine different experts (bottoms) by learnable gates. Because different experts can capture different knowledge which can relief part of the relationships sensitive problem among tasks.

3.2 Soft Parameter Sharing

Methods using soft parameter sharing (Lon, 2015; Ish, 2016; Dai et al., 2015; Yan, 2017) don't keep shared bottom layers. Besides, most of the model parameters are task-specific. Lon (2015) focuses on reducing the annotation effort of dependency parser tree. By combining two networks by L2 normalization mechanism, knowledge from a different source language can be use to reduce the requirement of amount of annotation. In some existing works, shallow layers will be separated to be regard as feature encoders to extract task-specific features. For example, Ish (2016) proposes Cross-Stitch model which is a typical separate bottom model. Cross-Stitch model will train on different task separately to encode task-specific knowledge into different bottom layers. A task-specific cross-stitch unit acts as gate to combine those separately trained layers together. Yan (2017), based on matrix factorization techniques used by conventional multi-task learning framework, introduces

tensor factorization model to allow common knowledge be shared at each layer in the network. By the strategy proposed in their work, parameters are softly shared across the corresponding layers of deep learning network and the parameter sharing ratio will be determined by model itself. The irrelevance of tasks problem will be relieved by this method in some extent.

Our work also falls into the soft parameter sharing category. We employ advanced neural network layer as feature encoder to extract task-specific feature. Also, like Ish (2016), interaction effects of multiple related tasks are involved into our model. In our work, we further address this problem from a fine-grained view. Our method try to not only model the high-level task dependency relationship like all existing works is doing. But also a data-dependent task dependency is also considered, which allow our method can leverage the hidden dependency relationship encoded in datasets more efficiently. Besides, none of the aforementioned works attempt to discover and construct interpretable dependency relationship, which is also one of the main contribution of this work.

4 APPROACH

In this section, we propose our framework 'Learning to Transfer via ModellIng mulTi-level Task dEpeNdency' (L2T-MITTEN), which can end-to-end learn the task dependency in both general and data-specific level, and help to ensemble multi-task representations from different task-specific bottom models.

4.1 PROBLEM FORMULATION

To formulate our framework, we first start by briefly introducing a general setting of multitask learning. For each task $t \in \{1, ..., T\}$, we have a corresponding dataset $D^{(t)} = \{(X_i^{(t)}, y_i^{(t)})\}_{i=1}^{N^{(t)}}$ with $N^{(t)}$ data samples, where $X_i^{(t)}$ represent the feature vector of i-th data sample and $y_i^{(t)}$ is the its label. We'd like to train T models for these tasks, each model has its own parameter $W^{(t)}$. Note that different multi-task learning framework, these parameters can be shared hardly or softly. The goal of multi-task learning is to minimize the normalized loss over all the T tasks by:

$$\min_{W} \sum_{t=1}^{T} \frac{\sum_{i=1}^{N^{(t)}} \mathcal{L}(W^{(t)}; X_i^{(t)}, y_i^{(t)})}{N^{(t)}} \tag{1}$$

4.2 ARCHITECTURE

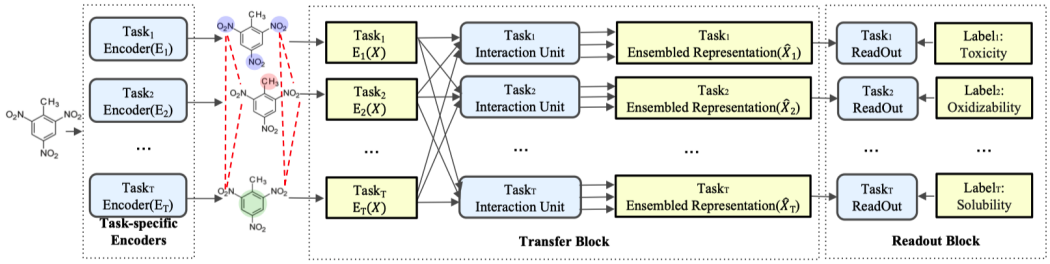


Figure 2: Overall architecture of our multi-task learning framework (L2T-MITTEN). For each position of the input data, we'll send its multi-task representations to each other, ensemble via a task-specific Interaction Unit to get output representation for each task.

As is shown in Figure 2, our framework consists of three components: Task-specific Encoders, Transfer Block, Readout Block.

The Task-specific Encoders is consisted by T separate feature encode, which can be any type feed-forward networks based on specific dataset. Unlike hard parameter sharing methods that tie the bottom encoders' parameters together, we keep each feature encoder separate to efficiently extract task specific knowledge. In this way, for a given training point X, we can use these encoders to get task-specific representations $\{E_t(X)\}_{t=1}^{T}$.

To conduct multi-task learning, we can simply use these the representation of each task alone to predict label without sharing any parameters. However, for task without sufficient labels, the model is not able to generalize well. Therefore, we'd like to transfer the knowledge among these T tasks by ensembling their representations, which is what we do in the transfer block.

In the transfer block, to bring task-specific features into together, we need to mutually map extracted feature of certain one task to the space of all the other ones. Therefore, there are total T^2 mapping functions are required. To prevent the quadratic growth of the number of mapping functions with the increasing of number of tasks, we decompose each mapping function into two separate parts. Assume we are trying to transfer extracted feature $E_i(X)$ to feature space of task j, where $i, j \in \{1, 2, ..., T\}$. We can decompose the mapping function $F_{ij}(\cdot)$ to $F_{Tj} \circ F_{Si}(\cdot)$. By this way, we only need $2T$ mapping functions in total for F_{Si} and F_{Tj}, where $i, j \in \{1, 2, ..., T\}$. The space complexity is reduced from $O(T^2)$ to $O(T)$. Here we denote the transferred representation as:

$$\boldsymbol{H}_{ij} = F_{Tj} \circ F_{Si}(E_i(\boldsymbol{X}_i)) \tag{2}$$

4.3 Modelling Multi-level Task Dependency

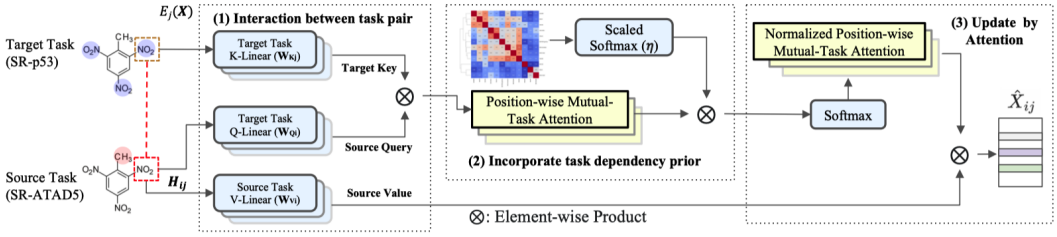


Figure 3: Position-wise Mutual Attention Mechanism

The key component of our framework is the Transfer Block, where we will model the multi-view task dependency, with which to ensemble representations from each source task. In here, the mutli-view task dependency is consisted by two parts: (1)the general task dependency and (2)the data-dependent task dependency. To model the general task dependency, we represent it by parameterized dependency graph $\boldsymbol{D}, \mathbb{R}^{T \times T}$. The learnable weight of this parameterized dependency graph represents the transferable weight of between any two tasks. Note that the dependency graph is asymmetrical. By this way, the negative influence from irrelevant tasks can be reduced as less as possible. Further, even for same task pair, the ratio of useful part is different among different data samples. Therefore, we study data-dependent task dependency in depth. Data-dependent task dependency is modeling by Position-wise Mutual Attention mechanism and general task dependency together. To efficiently model the data-dependent task dependency and reduce the irrelevant noise came from data instances, we only consider the mutual attention between representations of same data under different tasks. For example, certain node of one graph will only give attention to its counterpart under other different tasks. The detail of the Position-wise Mutual Attention Mechanism is shown in Figure 3.

Let us assume the representation of source task, $\boldsymbol{H}_{ij}$, is interacting with representation of target task representation $E_j(X_j)$, $i, j \in \{1, 2, ..., T\}$. We get the position-wise mutual attention by:

$$\boldsymbol{A}_{i \to j} = \sum_{i=0}^{d} \frac{(\boldsymbol{H}_{ij}\boldsymbol{W}_{Qi}) \otimes (E_j(\boldsymbol{X}_j)\boldsymbol{W}_{Kj})}{\sqrt{d}} \tag{3}$$

where d is the dimension of that representation, $\otimes$ is the Hadamard product.

$\boldsymbol{A}_{i \to j}$ is theposition-wise mutual attention between source task i and target task j. $\boldsymbol{W}_{Qi}, \boldsymbol{W}_{Kj}$ are query and key projection matrices. $\boldsymbol{W}_{Qi}, \boldsymbol{W}_{Kj} \in \mathbb{R}^{d \times d}$.

For certain target task j, there is a set of attention weights $\boldsymbol{A}_j = \{\boldsymbol{A}_{i \to j}\}, i \in \{1, 2, ..., T\}, i \neq j$, but each attention weight can only be aware of pairwise data dependency. To make data-dependent task dependence also take general task dependency into consideration, we scale the set of attention

$\boldsymbol{A}_j$ by parameterized dependency graph $\boldsymbol{D}$, and we normalize the dependency relation by $\tilde{\boldsymbol{D}} = \text{Softmax}(\boldsymbol{D})$. By weighted sum transferred representations according to the data dependency and task dependency, we can get the mixed representation of target task $\hat{\boldsymbol{X}}_j$:

$$\hat{\boldsymbol{X}}_j = \sum_{i=1}^{k} \left[\Big(\text{Softmax}(\eta \tilde{D}_{i \to j} \boldsymbol{A}_{i \to j}) \otimes (\boldsymbol{H}_{ij} \boldsymbol{W}_{vi}) \Big) \boldsymbol{W}_{vj} \right] \tag{4}$$

where, $\boldsymbol{W}_{vi}$ and $\boldsymbol{W}_{vj}$ are value projection matrices, $\boldsymbol{W}_{vi}, \boldsymbol{W}_{vj} \in \mathbb{R}^{d \times d}$, $\otimes$ is also the Hadamard product. And η is a scalar value which is to prevent the vanish gradient problem of two softmax operation.

5 EXPERIMENT

In this section, we evaluate the performance of our proposed L2T-MITTEN approach against a number of classical and SOTA approaches on two application domains: graph and text. In graph domain, we train a multitask Graph Convolutional Network (GCN) for both graph-level and node-level classification. And in text domain, we train a multitask Recurrent Neural Network (RNN) for text classification. Further, we provide visualization and analysis on the learned hidden dependency structure. Codes and datasets will be released.

5.1 DATASETS

For graph-level classification, we use Tox21 (Wu et al., 2017) and SIDER (Kuhn et al., 2010).

- **Tox21:** The Toxicology in the 21st Century (Tox21) is a database measuring toxicity of chemical compounds. This dataset contains qualitative toxicity assays for 8014 organic molecules on 12 different targets including nuclear receptors and stress response pathways. In our experiment, we treat each molecule as a graph and each toxicity assay as a binary graph-level classification task (for 12 tasks in total).
- **SIDER:** The Side Effect Resource (SIDER) is a database of marketed drugs and adverse drug reactions. This dataset contains qualitative drug side-effects measurements for 1427 drugs on 27 side-effects. In our experiment, we treat each drug (organic molecule) as a graph and the problem of predicting whether a given drug induces a side effect as a individual graph-level classification tasks (for 27 tasks in total).

For node-level classification, we use DBLP (Tang et al., 2008) and BlogCatalog (IV et al., 2009).

- **DBLP:** In the dataset, authors are represented by nodes and its feature is generated by titles of their papers. Two authors are linked together if they have co-authored at least two papers in 2014-2019. We uses 18 representative conferences as labels. An author is assigned to multiple labels if he/she has published papers in some conferences. The processed DBLP dataset is also published in our repository.
- **BlogCatalog:** The BlogCatalog is a collection of bloggers. In the dataset, bloggers are represeted as nodes, and there is a link between two bloggers if they are friends. The interests of each blogger can be tagged according to the categories that he/she published blogs in. This dataset uses 39 categories as labels. Each blogger is assigned to multiple labels if he/she published blog in some categories.

For text classification, we use TMDb[1] dataset.

- **TMDb:** The Movie Database (TMDb) dataset is a collection of information for 4803 movies. For each movie, the dataset includes information ranging from production company, production country, release date to plot, genre, popularity, etc. In our experiment, we select plots as the input with genres as the label. We treat the problem of predicting whether a given plot belongs to a genre as a individual text-level classification tasks (for 20 tasks in total).

[1]The Movie Database Website: https://www.themoviedb.org/.

A summary of the five datasets is provided in Appendix A.

5.2 BASELINE METHODS

We compare our L2T-MITTEN approach with both classical and SOTA approaches. The details are given as follows:

Single-task method:

- **Single-Task:** Simply train a network consists of encoder block and readout block for each task separately.

Classical multi-task method:

- **Shared-Bottom** (Caruana, 1997): This is a widely adopted multi-task learning framework which consists of a shared-bottom network (encoder block in our case) shared by all tasks, and a separate tower network (readout block in our case) for each specific task. The input is fed into the shared-bottom network, and the tower networks are built upon the output of the shared-bottom. Each tower will then produce the task-specific outputs.

SOTA multi-task methods:

- **Cross-Stitch** (Ish, 2016): This method uses a "cross-stitch" unit to learn the combination of shared and task-specific representation. The "cross-stitch" unit is a $k \times k$ trainable matrix (k is the number of tasks) which will transfer and fuse the representation among tasks by the following equation:

$$\begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_k \end{bmatrix} = \begin{bmatrix} \alpha_{11} & ... & \alpha_{1k} \\ \vdots & \ddots & \vdots \\ \alpha_{k1} & \dots & \alpha_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}$$

  where x_i is the output of the lower level layer for task i, α_{ij} is the transfer weight from task j to task i, and $\tilde{x}_i$ is the input of the higher level layer for task i.
- **MMoE** (Guo & Farooq, 2018): This method adopts the Multi-gate Mixture-of-Expert structure. This structure consists of multiple bottom networks (experts), and multiple gating networks which take the input features and output softmax gates assembling the experts with different weights. The assembled features are then passed into the task-specific tower networks.

All the baseline models use the same encoder and readout block for each task. The architecture details are provided in Appendix B.

5.3 EXPERIMENTAL SET-UP

We partition the datasets into 80:20 training/testing sets and evaluate our approach under multiple settings[2]: (1) Sufficient setting: all tasks have sufficient labeled training data; (2) Imbalanced setting: some tasks have more labeled training data than others; (3) Deficient setting: all tasks have deficient labeled training data. Models are trained for 100 epochs using the ADAM optimizer.

5.4 RESULT

We report the performance of our approach and baselines on graph classification, node classification and text classification tasks in terms of AUC-ROC score in Table 1 and 2 respectively.[3]

From the above result, first of all, we can see that multi-task methods outperform single-task method in most cases which shows the effectiveness of knowledge transfer and multi-task learning. Further,

[2]The settings are built by applying different masks to training sets, e.g. in imbalanced setting, we randomly mask some data samples in the training set.

[3]In the tables, "All $p\%$" means all tasks' labeled training data ratios are masked to $p\%$, while "Partially $p\%$" means only some randomly selected tasks are masked to $p\%$ labeled training data.

Table 1: Experiment result for graph datasets

Dataset	Labeled Data Ratio	Single-Task	Shared-Bottom	Cross-Stitch	MMoE	Our
Tox21	All 80%	0.8063	0.8171	0.8204	0.8049	**0.8333**
	Partially 10%	0.7138	0.7309	0.7128	**0.7331**	0.7310
	All 10%	0.7719	0.7934	0.7823	0.7848	**0.8033**
SIDER	All 80%	0.6458	0.6484	0.6676	0.6406	**0.6701**
	Partially 10%	**0.5682**	0.5534	0.5504	0.5377	0.5541
	All 10%	0.6277	0.6290	0.6285	0.6261	**0.6363**
DBLP	Partially 1%	0.8069	0.8056	0.5148	0.7930	**0.8241**
	All 10%	0.8232	0.8077	0.5150	0.8177	**0.8367**
BlogCatalog	Partially 5%	0.5154	0.6521	0.5259	0.6720	**0.6769**
	All 20%	0.6100	0.6667	0.5272	0.6850	**0.6861**

Table 2: Experiment result for text dataset

Dataset	Labeled Data Ratio	Single-Task	Shared-Bottom	Cross-Stitch	MMoE	Our
TMDb	All 80%	0.8172	0.8272	**0.8543**	0.8198	0.8484
	Partially 10%	0.7324	0.7293	0.7234	0.6590	**0.7404**
	All 10%	0.8033	0.8227	0.8452	0.7869	**0.8480**

we can see that our proposed L2T-MITTEN approach outperforms both classical and SOTA in most tasks. Finally, our approach shows significant improvement under deficient labeled training data setting, since our approach leverages the structure of data sample itself to guide the transfer among tasks.

Secondly, we found that in real-world dataset, like DBLP dataset, our model can outperform other SOTA methods significantly, which demonstrate that the importance of taking multi-view dependency into consideration. Note that the Single-Task can achieve the second best result. This fact indicates that in real-world dataset, tasks may be irrelevant with each other. Our multi-view dependency relationship can be more effective to prevent the influence of other irrelevant tasks.

Furthermore, we conduct experiments on text classification dataset, TMDb. The Cross-Stitch model achieve the best result when the label ratio for every task is 80%. However, our task can achieve best result for partially labeled setting(partially 10%) and few labeled setting(all 10%). This fact demonstrates that our directed task dependency graph can effectively prevent the negative knowledge be transferred among different tasks when training label is few.

5.5 Visualization and Analysis

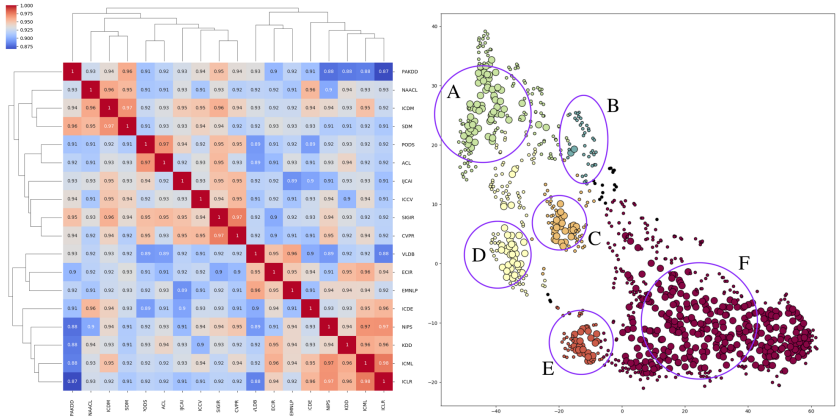
For visualization and analysis of the learned hidden dependency structure, we will take DBLP as an example here due to its simplicity in interpreting and understanding.

First, in Figure 4a, where we directly visualize $\boldsymbol{D}$, the learned task dependency matrix, we can see our approach indeed captures the task dependency structure in general, i.e. conferences from the same domain are more likely to be in the same sub-tree. Moreover, in Figure 4b we plot the authors (nodes) according to $\boldsymbol{DA}$, the learned data-specific task dependency matrix and we can see that there are some clusters formed by authors. Further, we visualize the mean value of $\boldsymbol{DA}$ for each cluster, as shown in Figure 4c. We can see that different cluster does have different task dependency, which is desirable, since when predicting if an author has published papers in some conferences, authors from different domains should have different transfer weight among conferences (tasks). As a summary, it is demonstrated that our approach is able to capture the task dependency at multiple level according to specific data.

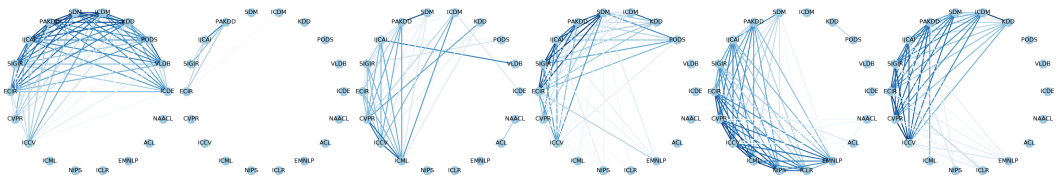
Figure 4: Visualization of the learned hidden dependency structure for DBLP dataset

(a) General tasks (conferences) dependency structure

(b) Nodes (authors) cluster



(c) Data-specific tasks (conferences) dependency structure for each cluster (Left: cluster A, Right: cluster F)



6 Conclusion

We proposed 'Learning to Transfer via ModellIng mulTi-level Task dEpeNdency' (L2T-MITTEN), a novel multi-task learning framework that efficiently extracts both general task dependency and data-dependent task dependency relationship, and uses it to enhance the performance on all tasks. L2T-MITTEN employs Position-wise Mutual Attention mechanism to capture the multi-view task dependency relationship and efficiently use the extracted dependency relationship to guide the inference. We design three experimental settings where training data is sufficient, imbalanced or deficient and validate our model extensively on multiple datasets against both classical and SOTA multitask methods. Experimental results demonstrate the superiority of our method against other baselines.

References

Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada, 2002. ACM. ISBN 1-58113-567-X.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers, 2015. The Association for Computer Linguistics. ISBN 978-1-941643-73-0. URL https://www.aclweb.org/anthology/volumes/P15-2/.

2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016. IEEE Computer Society. ISBN 978-1-4673-8851-1. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7776647.

5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017. OpenReview.net. URL https://openreview.net/group?id=ICLR.cc/2017/conference.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Jonathan Baxter. A model of inductive bias learning. *CoRR*, abs/1106.0245, 2011. URL http://arxiv.org/abs/1106.0245.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. doi: 10.1023/A:1007379606734. URL https://doi.org/10.1023/A:1007379606734.

Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.). *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015. URL http://papers.nips.cc/book/advances-in-neural-information-processing-systems-28-2015.

Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. *CoRR*, abs/1512.04412, 2015. URL http://arxiv.org/abs/1512.04412.

Jennifer G. Dy and Andreas Krause (eds.). *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 2018. PMLR. URL http://proceedings.mlr.press/v80/.

Yike Guo and Faisal Farooq (eds.). *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2018. ACM. doi: 10.1145/3219819. URL https://doi.org/10.1145/3219819.

John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki (eds.). *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, 2009. ACM. ISBN 978-1-60558-495-9.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

Michael Kuhn, Monica Campillos, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. A side effect resource to capture phenotypic effects of drugs. *Molecular Systems Biology*, 6(1):343, 2010. doi: 10.1038/msb.2009.98. URL https://www.embopress.org/doi/abs/10.1038/msb.2009.98.

Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.). *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017. Association for Computational Linguistics. ISBN 978-1-945626-83-8. URL https://www.aclweb.org/anthology/volumes/D17-1/.

Bharath Ramsundar, Steven M. Kearnes, Patrick Riley, Dale Webster, David E. Konerding, and Vijay S. Pande. Massively multitask networks for drug discovery. *CoRR*, abs/1502.02072, 2015. URL http://arxiv.org/abs/1502.02072.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017. URL http://arxiv.org/abs/1706.05098.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *CoRR*, abs/1810.04650, 2018. URL http://arxiv.org/abs/1810.04650.

Trevor Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *CoRR*, abs/1905.07553, 2019. URL http://arxiv.org/abs/1905.07553.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pp. 990–998, 2008.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A Benchmark for Molecular Machine Learning. *arXiv e-prints*, art. arXiv:1703.00564, Mar 2017.

A Dataset Summary

Table 3: Graph datasets summary

Dataset	Source	Graphs	Nodes Avg.	Edges Avg.	Graph Labels	Node Labels
Tox21	Bio	8014	18	48	12	-
SIDER	Bio	1427	33	105	27	-
DBLP	Citation	1	14704	24778	-	18
BlogCatalog	Social	1	10312	333983	-	39

Table 4: Text dataset summary

Dataset	Movies	Words Avg.	Genres
TMDb	8014	59	20

B Architecture Details

B.1 Architecture Details for Graph Model

As shown in Figure 5, in the Encoder Block, we use several layers of graph convolutional layers (Kipf & Welling, 2016) followed by the layer normalization (Ba et al., 2016). In the Readout Block, for graph-level task, we use set-to-set (Vinyals et al., 2015) as the global pooling operator to extract the graph-level representation which is later fed to a classifier; while for node-level task, we simply eliminate the global pooling layer and feed the node-level representation directly to the classifier.

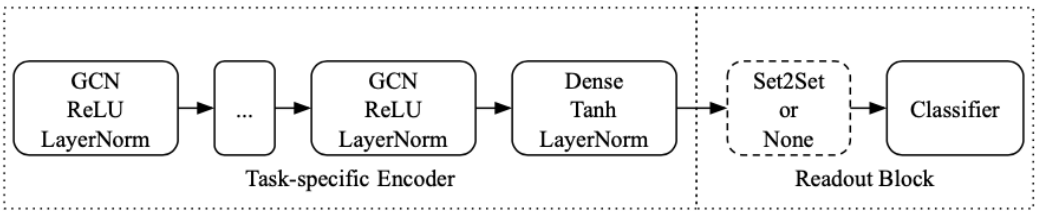


Figure 5: Graph convolutional networks architecture. Note that in node-level task, the Set2Set layer (global pooling) is eliminated.

B.2 Architecture Details for Text Model

The text model uses long short-term memory (LSTM) architecture in their Encoder Block, and the dot-product attention in the Readout Block, as shown in Figure 6. The dot-product attention used to get the text-level representation is as follows:

$$\boldsymbol{\alpha} = \text{Softmax}(\boldsymbol{O}\boldsymbol{H}_n^T); \quad \hat{\boldsymbol{O}} = \boldsymbol{\alpha}^T\boldsymbol{O}$$

where $\boldsymbol{O} \in \mathbb{R}^{n\times d}$ is the output of the LSTM, $\boldsymbol{H_n} \in \mathbb{R}^{1\times d}$ is the hidden state for the last word, $\boldsymbol{\alpha} \in \mathbb{R}^{n\times 1}$ is attention weight for each word, and $\hat{\boldsymbol{O}} \in \mathbb{R}^{1\times d}$ is the text-level representation (n is the number of words, d is the feature dimension for each word).

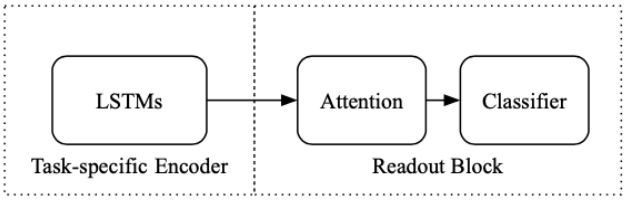


Figure 6: Text classification network architecture.