MULTI-RELATION MESSAGE PASSING FOR MULTI-LABEL TEXT CLASSIFICATION

Muberra Ozmen* *Hao Zhang*† *Pengyun Wang*‡ *Mark Coates**

* McGill University, Montreal, QC, Canada
† Hong Kong University of Science and Technology, Hong Kong
‡ Huawei Noah's Ark Lab, Shenzhen, China

ABSTRACT

A well-known challenge associated with the multi-label classification problem is modelling dependencies between labels. Most attempts at modelling label dependencies focus on co-occurrences, ignoring the valuable information that can be extracted by detecting label subsets that rarely occur together. For example, consider customer product reviews; a product probably would not simultaneously be tagged by both "recommended" (i.e., reviewer is happy and recommends the product) and "urgent" (i.e., the review suggests immediate action to remedy an unsatisfactory experience). Aside from the consideration of positive and negative dependencies, the direction of a relationship should also be considered. For a multi-label image classification problem, the "ship" and "sea" labels have an obvious dependency, but the presence of the former implies the latter much more strongly than the other way around. These examples motivate the modelling of multiple types of bi-directional relationships between labels. In this paper, we propose a novel method, entitled Multi-relation Message Passing (MrMP), for the multi-label classification problem. Experiments on benchmark multi-label text classification datasets show that the MrMP module yields similar or superior performance compared to state-of-the-art methods. The approach imposes only minor additional computational and memory overheads.[1]

Index Terms— multi-label classification, text classification, multi-relation GNNs

1. INTRODUCTION

Multi-label classification involves selecting the correct subset of tags for each instance from the available label set. There are numerous real world applications ranging from image annotation to document categorization [1]. A naive approach consists of converting the problem into multiple binary classification problems. This is the *binary relevance (BR)* method [2]. A critical difference between multi-label and multi-class classification is that the class values are not mutually exclusive in multi-label learning. Usually, we anticipate that there are dependencies between the labels, and there are often multiple different types of relationships. Incorporating and learning the dependency relationships between labels during the learning process has an appealing potential of boosting predictive performance.

The simplest approach that considers label dependencies is the *label powerset (LP)* method [3]. This involves converting the problem to multi-class classification by treating each possible combination of labels as a separate class, i.e., a multi-label problem with L possible labels would be converted to a multi-class problem with 2^L classes. While the LP method enables us to make use of powerful multi-class classification methods, it suffers from scaling to problems with a large number of labels. Aside from the LP method, the existing methods that account for label dependencies construct the relationships between labels by assessing co-occurrences in the training data [4, 5, 6]. Co-occurrences represent a *pulling* type of relationship between labels (labels that often appear together should be strongly encouraged to appear together in predictions). However, focusing solely on frequent co-occurrences ignores other valuable information. In particular, there should be a distinction between labels that occasionally appear together and those that *never* appear together. Aside from statistical label dependencies determined by co-occurrence (or the lack thereof), there are other valuable relations that should be considered when designing a classifier. These include semantic relations (e.g., synonyms, plural forms) or predefined structural relations (e.g., hierarchical category relations).

In this paper, we propose a novel method, called *Multi-relation Message Passing (MrMP)*, which employs a *Compositional Graph Convolutional Network (CompGCN)* [7] to model multiple bi-directional relationships between labels. We make the following contributions:

1. We design a multi-relation label embedding module that can be integrated into most encoder-decoder type neural network architectures to account for the dependencies between labels.
2. We propose a simple and efficient method to extract statistically significant *pulling* and *pushing* relations between labels.
3. We design a *Transformer* [8] based message passing network for the multi-label text classification problem considering two types of statistical relations between labels.
4. We perform experiments on benchmark multi-label text classification datasets, comparing with classical and state-of-the-art baselines, to demonstrate the efficacy of our method.

2. RELATED WORK

There have been multiple attempts in the literature to capture label dependencies for multi-label classification problems. *Probabilistic classifier chains (PCC)* stack a sequence of binary classifiers and predict one label at a time conditioned on previously predicted labels [9, 10, 11]. PCC based methods decompose the joint probability of observing label subsets into a product of conditional probabilities. The computational complexity therefore increases exponentially with the number of possible labels. To reduce the length of the classifier chain and the corresponding model complexity, [12] suggests *ML-RNN*, a sequence-to-sequence architecture that focuses on predicting positive labels only. PCC models can suffer from the sub-optimal pre-defined label ordering. Training and inference time

[1] The implementation of our method is publicly available at `https://github.com/muberraozmen/MrMP`.

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number 260250]

is an issue due to the inability to harness parallel computation.

In latent embedding learning methods the inputs and outputs are projected into a shared latent space [13, 14, 15]. An effective recent method is the *Multivariate Probit Variational AutoEncoder (MPVAE)* [6]. This maps features and labels to probabilistic subspaces and uses the Multivariate Probit (MP) model to make predictions by sampling from these subspaces. A shared covariance matrix is learned to capture label dependencies. However, the embedded learning methods fail to incorporate prior knowledge about label structures.

Graph-based methods for capturing label dependencies have shown promising performance [4, 16]. [4] use *Graph Convolutional Networks (GCNs)* [17] to map label representations to interdependent object classifiers for the multi-label image classification task. In these methods, a label graph is typically built based on label co-occurrence, with nodes corresponding to labels and edges corresponding to how two labels interact. The label graph can be combined with graph neural networks to enable the interactive learning of features and label embeddings. [5] and [18] propose starting with a simple fully-connected graph or a co-occurrence label graph. A *Message Passing Neural Network (MPNN)* then passes messages among label embeddings and features to enable learning of high order label correlation structure that is conditioned on the features. Due to the greater flexibility offered by MPNNs compared to GCNs, these models achieve state-of-the-art performance on many benchmark datasets. However, all of the above graph methods use simple undirected graphs, neglecting the richness of information present in the joint empirical probability distribution of the labels, such as whether the occurrence of label A encourages or suppresses the occurrence of label B, relative to the marginal distribution of the latter.

This work proposes the construction of a multi-relational label graph and uses multi-relational graph neural networks to more comprehensively capture the information present in the joint empirical probability distribution of the labels. The inclusion of the richer information in the model can potentially improve performance for multi-label classification problems.

3. MULTI-RELATION MESSAGE PASSING (MRMP)

For the multi-label classification problem, the advantage of using Message Passing Neural Networks (MPNNs) arises due to the structural power of graphs for representing global label dependencies. The use of RNN based architectures has two often overlooked advantages. The ordering in the prediction path allows the architecture to focus on the next most probable label and ignore irrelevant labels. Moreover, the ordering models dependencies in a directed manner, albeit in only one direction. Our proposed method strives to combine these advantages. The proposed model contains an encoder for extracting features and a decoder for predicting labels. Although the method can be applied to a variety of types of input datasets, we focus on the input type of text to provide a more concrete description.

3.1. Notation and Preliminaries

We denote a sample by a sequence of words $\boldsymbol{x}$ which is composed of N components $\boldsymbol{x} = (x_1, x_2, ..., x_N)$ where N is the length of sequence. Associated with each sample is its corresponding label set, represented as a binary vector $\boldsymbol{y} = (y_1, y_2, \dots, y_L)$, where $y_i \in \{0, 1\}$ indicates whether label i appears. We use a training set $(\boldsymbol{x}_j, \boldsymbol{y}_j)_{j\in\mathcal{T}}$ to learn the parameters of a predictive model. The model then takes $\boldsymbol{x}$ as an input and outputs $\hat{\boldsymbol{y}} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_L\}$ where $0 \leq \hat{y}_i \leq 1$ is the estimated probability that label i appears. We denote input feature embeddings by $\boldsymbol{z} = \{z_1, ..., z_S\}$ and label specific feature embeddings by $\boldsymbol{u} = \{u_1, ..., u_L\}$. The estimations are evaluated by mean binary cross entropy $\mathcal{L}_{\text{bce}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{L}\sum_{i=1}^{L} -y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$.

Multi-head Attention (MHA) is performed by concatenating parallel heads of scaled dot-product attention. The main motivation is to prevent destabilization during training by bad initialization of learnable weights. The inputs are generalized as $\boldsymbol{Q}$ for "Query", $\boldsymbol{K}$ for "Keys", and $\boldsymbol{V}$ for "Value". Scaled-dot product attention is formulated as follows:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V}\,. \tag{1}$$

Positionwise Feed Forward (PFF) networks are fully connected feed-forward network layers which consist of two linear transformations, with a ReLU activation between them:

$$\text{PFF}(\boldsymbol{x}) = \max(0, \boldsymbol{x}\boldsymbol{W}_1 + b_1)\boldsymbol{W}_2 + b_2\,. \tag{2}$$

3.2. Proposed Model

We consider two types of statistical relations between labels; *pulling* and *pushing*. Let a relation graph be represented by adjacency matrices $\boldsymbol{A}^+$ and $\boldsymbol{A}^-$ for pulling and pushing edges, respectively. The method used to calculate the corresponding adjacency matrices will be explained later. To properly utilize the multi-relational adjacency matrices obtained, we propose a Multi-relation Message Passing (MrMP) method. An overview of the model is provided in Figure 1. In essence, it combines the ideas of LaMP [5] and CompGCN [7]. Assume that the input is a sequence of tokens $\boldsymbol{x} = \{x_1, x_2, ..., x_N\}$. The encoder initializes input word embeddings and applies positional encoding if the data type is sequential [8], then stacks several layers of self-attention to construct the sequential representation of input at each layer, $\boldsymbol{z}^l = \{z_1^l, z_2^l, ..., z_S^l\}$. The decoder is composed of a label-relation module and a label-feature module. The label-relation module is responsible for capturing the multi-relational dependencies among labels, while the label-feature module extracts the most relevant features for each label.

The label-relation module initializes the label node embeddings and stacks CompGCN layers with pulling and pushing relations. Let z_+ and z_- represent pulling and pushing relation embeddings, respectively. Let $\boldsymbol{W}_+$ and $\boldsymbol{W}_-$ be trainable weight matrices for pulling and pushing relations, respectively. At each layer of the label relation module the label embeddings $\boldsymbol{V}^{l+1} = [v_1^{l+1}, v_2^{l+1}, ..., v_L^{l+1}]$ are updated based on the output from the previous layer $\boldsymbol{V}^l$ as follows:

$$v_i^{l+1} = f\Bigg(\sum_{j\in\mathcal{N}^+(i)} \boldsymbol{W}_+^l \phi(v_i^l, z_+^l) + \sum_{j\in\mathcal{N}^-(i)} \boldsymbol{W}_-^l \phi(v_i^l, z_-^l)\Bigg)\,. \tag{3}$$

Here the neighbourhoods for each relation, $\mathcal{N}^+(i)$ and $\mathcal{N}^-(i)$, include the original relations, inverse relations and self loops. In our experiments we set $\phi(\cdot, \cdot)$, the composition function that combines the label hidden state and the relation hidden state, to be summation. In our case, since we also have an intuitive understanding of the relationship between the two relations, we set the relation embeddings to be $z_+^l = -z_-^l$ at each layer. The relation embeddings are updated at each layer by relation specific trainable parameters, $z_+^{l+1} = \boldsymbol{W}_{rel} z_+^l$.

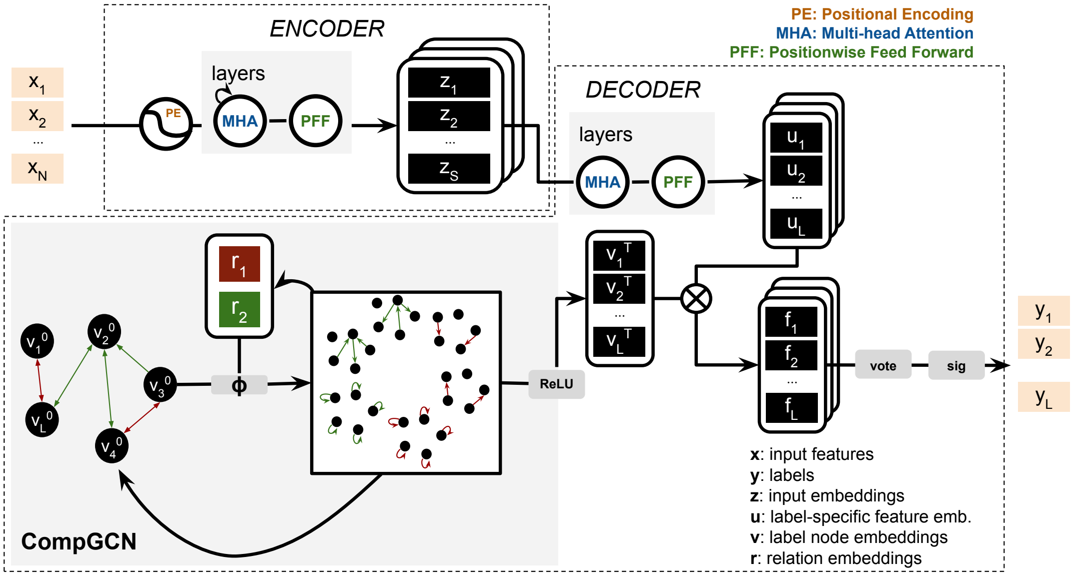

Fig. 1. Overview of MrMP. The encoder first applies positional embedding on the input sequence $\boldsymbol{x}$. The output then passes through encoder-self attention layers of MHA and a PFF to obtain the input word embeddings $\boldsymbol{z}$. The decoder initializes label $\boldsymbol{v}^{\mathbf{0}}$ and relation embeddings $\boldsymbol{r}$ and composes them together using a function ϕ to prepare the multi-relation layers' input. The layers of the multi-relation module aggregate the label embeddings, updated by **pulling** and **pushing** relation sets, and compose them with updated relation embeddings recursively to feed the next layer. The label node embeddings $\boldsymbol{v}^{\boldsymbol{T}}$ are obtained by applying a ReLU on the final layer's output. The encoder-decoder attention layers of MHA and PFF prepare label-specific feature embeddings $\boldsymbol{u}$. $\boldsymbol{f}$ is obtained by element-wise product of $\boldsymbol{v}$ and $\boldsymbol{u}$. The label predictions $\hat{\boldsymbol{y}}$ are obtained after voting on $\boldsymbol{f}$ caused by multiple layers of encoder output and applying a sigmoid to normalize.

We use the final label embeddings $\boldsymbol{V}^T$ as queries to extract label-specific features $\boldsymbol{U}$ based on the output from the encoder. Since different labels may be most effectively determined by features at different layers, we extract label specific features $\boldsymbol{U}^l$ based on each $\boldsymbol{z}^l$, respectively. On a given decoder layer the label specific feature embeddings are updated as follows:

$$m_i^l = u_i^l + \sum_{j=1}^{S} \text{MHA}(u_i^l, z_j^l; \boldsymbol{W}_r^l), \tag{4}$$

$$u_i^{l+1} = m_i^l + \text{PFF}(m_i^l; \boldsymbol{W}_r)\,, \tag{5}$$

where $u_i^0 = v_i^T$. Then, label prediction is performed based on each $\boldsymbol{U}^l$ to produce $\hat{\boldsymbol{f}}^l = \text{diag}(\boldsymbol{U}^l * \boldsymbol{V}^T)$ where $*$ stands for element-wise multiplication. Finally, the ultimate prediction for class probabilities $\hat{\mathbf{y}}$ is determined by a voting, $\hat{\boldsymbol{y}} = \sum_l \hat{\boldsymbol{f}}^l$.

Calculating Label Relation Graphs. The conventional approach for calculation of the prior graph that represents the relationship between labels involves estimating conditional probabilities based on co-occurrences in the training data. LaMP (with prior) [5] uses a simpler method that forms an edge between two labels if they co-occur in any training sample. We consider two types of statistical relations between labels: pulling and pushing. We first test the existence of a relation based on occurrences for each label pair, and then decide whether that relation is pushing or pulling. That is, given the presence of dependency between two labels, we hypothesize whether it corresponds to co-occurrence or avoidance relation. For all label pairs i and j,

1. Perform a dependence test by setting null hypothesis as H_0 : $P(L_j = 1|L_i = 1) \neq P(L_j = 1)$. We test this via a chi-squared test statistic on a pairwise contingency table.
2. Determine type of relation: If the label pair passes the hypothesis test, the existence of label j significantly depends on the existence of label i. The label pair is set to have a pulling edge if $P(L_j = 1|L_i = 1) > P(L_j = 1)$, and a pushing edge otherwise.

Relational Loss Function. We use a combination of cross-entropy loss and a relation based label embedding distance for training our model. Let the final label node embedding for label i be denoted by $\hat{v}_i$. The relation based loss is formulated as follows:

$$\mathcal{L}_{\text{rel}}(\hat{\boldsymbol{v}}_i) = -\frac{1}{|\mathcal{N}^+(i)|} \sum_{j \in \mathcal{N}^+(i)} \frac{\hat{\boldsymbol{v}}_i . \hat{\boldsymbol{v}}_j}{\|\hat{\boldsymbol{v}}_i\| \|\hat{\boldsymbol{v}}_j\|} + \frac{1}{|\mathcal{N}^-(i)|} \sum_{j \in \mathcal{N}^-(i)} \frac{\hat{\boldsymbol{v}}_i . \hat{\boldsymbol{v}}_j}{\|\hat{\boldsymbol{v}}_i\| \|\hat{\boldsymbol{v}}_j\|}\,. \tag{6}$$

4. EXPERIMENTS

4.1. Datasets and Evaluation Metrics

Although the proposed method can be extended to different domains, currently we have experimented with 4 benchmark multi-label text classification datasets. Bibtex, Bookmarks [19] and Delicious [20] involve automated tag suggestion for entries from the BibSonomy social publication, the Bookmark sharing system, and for web pages from the del.ico.us social bookmarking site, respectively. Reuters-21578 [21] is a collection of newswire stories. Our experiments cover a range of dataset sizes between 7,538 and 804,410. The number of labels range between 90 and 983; and vocabulary size ranges between 500 and 50,000. We use the same data splits as described in [5] for fair comparison.

Table 1. Experiment Results

	Bibtex				**Bookmarks**				**Delicious**				**Reuters**			
	ACC	ebF1	maF1	miF1	ACC	ebF1	maF1	miF1	ACC	ebF1	maF1	miF1	ACC	ebF1	maF1	miF1
ML-KNN	0.074	0.205	0.134	0.268	0.202	0.241	0.133	0.264	0.003	0.223	0.080	0.245	0.651	0.710	0.253	0.732
ML-ARAM	0.107	0.258	0.097	0.238	0.181	0.236	0.127	0.184	0.005	0.155	0.041	0.167	0.474	0.673	0.163	0.626
ML-RNN	-	-	-	-	-	-	-	-	-	-	-	-	0.828	0.894	0.457	0.858
LaMP	0.185	0.447	0.376	0.473	0.242	0.389	0.286	0.373	*0.006*	*0.372*	*0.196*	*0.386*	0.835	0.906	0.560	0.889
MPVAE	*0.179*	0.453	0.386	0.475	*0.234*	*0.390*	***0.295***	*0.378*	*0.000*	0.373	0.181	0.384	*0.816*	*0.898*	*0.542*	*0.887*
MrMP	**0.199**	**0.460**	**0.393**	**0.481**	**0.251**	**0.397**	0.293	**0.384**	**0.007**	**0.377**	**0.199**	**0.391**	**0.844**	**0.914**	**0.591**	**0.893**
%	7.46%	1.54%	1.74%	1.26%	3.63%	1.89%	-0.70%	1.64%	15.12%	1.09%	1.42%	1.17%	1.04%	0.91%	5.60%	0.42%

The instance-based performance metrics we use to evaluate our method are example based F1 score (ebF1), and subset accuracy (ACC). The label-based performance metrics are micro-averaged F1 score (miF1), and macro-averaged F1 score (maF1). Subset accuracy measures the fraction of times that an algorithm identifies the correct subset of labels for each instance. The example-based F1 score is aggregated over samples and the macro-averaged F1 score over labels. The micro-averaged F1 score takes the average of the F1 score weighted by the contribution of each label, and thus takes label imbalance into account.

4.2. Baselines

In addition to state-of-art methods ML-RNN [12], LaMP [5], and MPVAE [6], we compare our algorithm to baseline classifiers such as ML-KNN [22] and ML-ARAM [23]. ML-KNN finds the nearest examples to a test class by the k-Nearest Neighbors algorithm and selects assigned labels using Bayesian inference. ML-ARAM use Adaptive Resonance Theory (ART) based clustering and Bayesian inference to calculate label probabilities. The results for ML-KNN and ML-ARAM are obtained by functions provided in the scikit-learn library [24]. For ML-RNN and LaMP we provide the results reported in the corresponding papers. For MPVAE we have reproduced the results in order to obtain performance metrics for the datasets under study. We provide the reported result if it exists and is better than those we reproduced.[2]

4.3. Implementation Details

We follow [5] for setting the relevant hyperparameters. For all datasets, the latent model dimensionality of the neural network is set to 512, the number of encoder and decoder layers to 2, and the number of attention heads to 4. The dropout probability is 0.2 for Bibtex and Reuters, and 0.1 for the other datasets. The significance level of the dependency test to calculate label graphs' adjacency matrices is selected to be 0.05. The value of the threshold used to convert soft prediction to predicted class is determined using the validation sets and optimized individually for all performance metrics. The model is trained with a batch size of 32, and the Adam [25] optimizer is used to compute gradients and update parameters with an initial learning rate of 0.0002 and step size 10 for 10% decay rate.

4.4. Results

The results are provided in Table 1. Overall our experiments show that Multi-relation Message Passing (MrMP) yields better performance than the state-of-art methods most of the time. The value of the multi-relation approach is most evident in subset accuracy in which MrMP outperforms the baselines by %7 on average. The overall performance improvements for the ebF1, miF1 and maF1 metrics are also positive, but below 2%. The maF1 improvement is slightly greater than the miF1 improvement. Comparatively, maF1 focuses more on rare labels; this is in line with the intuition that the guidance provided by incorporating multiple relations is more useful for labels with low observation frequency.

ML-KNN and ML-ARAM. MrMP yields better results than ML-KNN and ML-ARAM for all metrics since these two algorithms rely only on clustering of input word embeddings for mapping features to labels, while MrMP employs a message passing neural network where input word embeddings are mapped to label embeddings via multi-head attention.

ML-RNN. By following a prediction path, ML-RNN is able to focus on the next probable label given a previously predicted set of labels. This enhances the capability of capturing the full label subset for each instance, so ML-RNN is most competitive for the subset accuracy metric. The results on Reuters-21578 show that MrMP outperforms ML-RNN by 5% and 14% in miF1 and maF1 while achieving slightly better subset accuracy. The integration of the MrMP module allows the architecture to emulate the advantage of making predictions sequentially, while enjoying the flexibility and computational efficiency of a MPNN.

LaMP. Comparison to LaMP reveals the true impact of the multi-relation approach in terms of capturing label dependencies. While our model is most similar to LaMP in terms of the main building blocks of mapping features to labels, we employ multi-relation GCNs to model label dependencies instead of decoder self attention as in LaMP. The enhancement of subset accuracy performance compared to LaMP reveals the importance of accounting for the pushing type of relation between labels.

MPVAE. The comparisons with MPVAE show that multi-head attention based mapping of input to output and label dependency modelling are more effective than using a multivariate probit model for the mapping and a global covariance matrix for dependency modelling.

5. CONCLUSION

In this paper, we propose a Multi-relation Message Passing Network (MrMP) for the multi-label classification problem. The proposed model can be adapted to any number of relations as long as the prior is known. We consider *pulling* and *pushing* statistical relations between labels and use the Composition-based GCN to learn label embeddings that reflect these statistical relations. The experiments show that MrMP outperforms state-of-art models in terms of effectively modelling label dependencies in order to achieve improved classification performance.

[2] The reproduced results of MPVAE are in italic font in the tables.

6. REFERENCES

[1] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Deroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45, no. 9, pp. 3084–3104, 2012.

[2] O. Luaces, J. Díez, J. Barranquero, J. del Coz, and A. Bahamonde, "Binary relevance efficacy for multi-label classification," *Prog. Artificial Intell.*, p. 303–313, 2012.

[3] G. Tsoumakas and I. Katakis, "Multi-label classification: an overview," *Int. J. Data Warehousing and Mining*, vol. 3, pp. 1–13, 2007.

[4] Z. Chen, X. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2019.

[5] J. Lanchantin, A. Sekhon, and Y. Qi, "Neural message passing for multi-label classification," in *Proc. ECML-PKDD*, 2020, vol. 11907, pp. 138–163.

[6] J. Bai, S. Kong, and C. Gomes, "Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020.

[7] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *Proc. Int. Conf. Learning Rep.*, 2020.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[9] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learn.*, vol. 85, no. 3, pp. 333–359, 2011.

[10] K. Dembczynski, W. Cheng, and E. Hüllermeier, "Bayes optimal multi-label classification via probabilistic classifier chains," in *Proc. Int. Conf. Machine Learn.*, 2010, p. 279–286.

[11] R. Senge, J. José del Coz, and E. Hüllermeier, "Rectifying classifier chains for multi-label classification," *arXiv preprint arXiv:1906.02915*, 2019.

[12] J. Nam, E. Loza Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5413–5423.

[13] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, p. 730–738.

[14] C. Chen, H. Wang, W. Liu, X. Zhao, T. Hu, and G. Chen, "Two-stage label embedding via neural factorization machine for multi-label classification," in *Proc. AAAI Conf. Artificial Intell.*, 2019, pp. 3304–3311.

[15] J. Ma, B. Chi Y. Chiu, and T. W. S. Chow, "Multi-label classification with group-based mapping: A framework with local feature selection and local label correlation," *IEEE Trans. Cybernetics*, pp. 1–15, 2020.

[16] Y. Wang, D. He, F. Li, X. Long, Z. Zhou, J. Ma, and S. Wen, "Multi-label classification with label graph superimposing," *arXiv preprint arXiv:1911.09243*, 2019.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learning Rep.*, 2017.

[18] W. Zhao, S. Kong, J. Bai, D. Fink, and C. Gomes, "Hot-vae: Learning high-order label correlation for multi-label classification via attention-based variational autoencoders," *arXiv preprint arXiv:2103.06375*, 2021.

[19] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multi-label text classification for automated tag suggestion," in *Proc. ECML-PKDD Discovery Challenge*, 2008.

[20] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multi-label classification in domains with large number of labels," in *Proc. ECML-PKDD Workshop on Mining Multi-dimensional Data*, 2008.

[21] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, 2004.

[22] M. Zhang and Z. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[23] F. Benites and E. Sapozhnikova, "Haram: A hierarchical aram neural network for large-scale text classification," in *Proc. IEEE Int. Conf. Data Mining Workshop*, 2015, pp. 847–854.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Rep.*, 2015.