Hierarchical Transfer Learning for Multi-label Text Classification

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, Kostas Tsioutsiouliklis
Yahoo Research
701 First Avenue
Sunnyvale, CA, USA
{siddb, cakkaya, fperez, kostas}@verizonmedia.com

Abstract

Multi-Label Hierarchical Text Classification (MLHTC) is the task of categorizing documents into one or more topics organized in an hierarchical taxonomy. MLHTC can be formulated by combining multiple binary classification problems with an independent classifier for each category. We propose a novel transfer learning based strategy, HTrans, where binary classifiers at lower levels in the hierarchy are initialized using parameters of the parent classifier and fine-tuned on the child category classification task. In HTrans, we use a Gated Recurrent Unit (GRU)-based deep learning architecture coupled with attention. Compared to binary classifiers trained from scratch, our HTrans approach results in significant improvements of 1% on micro-F1 and 3% on macro-F1 on the RCV1 dataset. Our experiments also show that binary classifiers trained from scratch are significantly better than single multi-label models.

1 Introduction

Two main approaches for Multi-Label Hierarchical Text Classification (MLHTC) have been proposed (Tsoumakas and Katakis, 2007): 1. transforming the problem to a collection of independent binary classification problems by training a classifier for each category 2. training a single multi-label model that can predict all categories for instances simultaneously.

In a hierarchical taxonomy of categories, dependencies exist between parent and child categories that should be exploited when training classifiers. Recent work on MLHTC uses a Deep Graph-based Convolutional Neural Network (DGCNN) (Peng et al., 2018) -based single multi-label model with a recursive regularization component to model dependencies between parent and child categories. However, multi-label models suffer on categories with very few training examples (Krawczyk, 2016) due to data imbalance. Due to a large prediction space (all categories) of multi-label models, it is very difficult to optimize class weights to handle data imbalance. By contrast, binary classifiers provide more flexibility as class weights for each classifier can easily be optimized based on validation metrics. With a reasonable number of categories (few hundreds), collection of binary classifiers are a feasible option to solve MLHTC problems.

Influenced by recent progress of transfer learning on Natural Language Processing (NLP) tasks (Howard and Ruder, 2018; Mou et al., 2016), we present **HTrans**, a **H**ierarchical **Trans**fer Learning approach. We hypothesize that introducing dependencies between parent and child categories is possible using transfer learning. Therefore, we initialize parameters of the child category classifier from the binary parent category classifier and later fine-tune the model. The transfer of parameters can provide a better starting point for the child category classifier than training from scratch using randomly initialized parameters. Without any loss of generality, we propose a simple classification model using Gated Recurrent Unit (GRU) (Cho et al., 2014) coupled with attention (Dzmitry et al., 2015). We also select optimal class weights for each category to account for class imbalance (Burez and Van den Poel, 2009) in the data.

Our experiments on the RCV1 (Lewis et al., 2004) dataset show that **HTrans** improves over training models from scratch by 1% and 3% on micro-F1 and macro-F1 scores, respectively. Furthermore, we also show that binary models based on our architecture surpass DGCNN (state-of-the-art multi label model on RCV1 dataset) by 4% and 19% on micro-F1 and macro-F1 scores, respectively. Class weight optimization in itself produces

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6295–6300
Florence, Italy, July 28 - August 2, 2019.

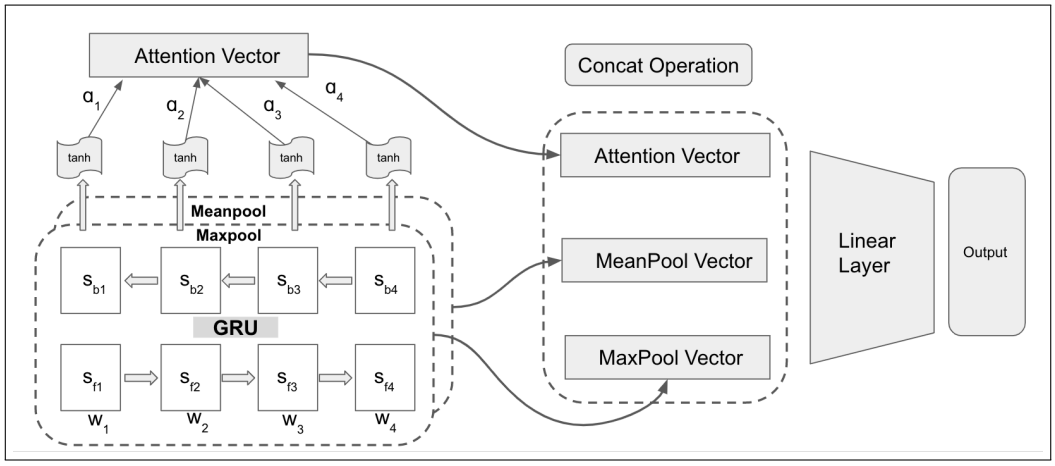

Figure 1: Architecture of our Proposed Model

an improvement of ∼9% on macro-F1 scores.

2 Related Work

A major focus of multi-label text classification research has been exploiting possible label dependencies to improve predictive performance. To account for label dependencies, some approaches utilize label correlations found in the training data (Tsoumakas et al., 2009; Huang and Zhou, 2012; Zhang and Zhang, 2010; Guo and Gu, 2011). Others make use of pre-defined label hierarchies. These approaches usually employ hierarchy-induced model regularization by putting constraints on the weight vectors of adjacent models, a type of transfer learning (Zhou et al., 2011; Gopal and Yang, 2013; Peng et al., 2018). **HTrans** is similar to the latter category of work as it uses transfer learning. We utilize fine-tuning to introduce inductive bias from a parent category to its children, whereas previous approaches use model regularization. Results are compared to the state-of-the-art DGCNN (Peng et al., 2018) model where a graph-based Convolutional neural network model is deployed in combination with recursive model regularization.

Fine-tuning of pre-trained models has shown promising results on various NLP tasks. Some of these approaches employ supervised pre-training transferring knowledge between related tasks (Mou et al., 2016; Min et al., 2017; Conneau et al., 2017). Another set of research focuses on a more general transfer task where models are pre-trained on a language modeling task on large unsupervised corpora and later fine-tuned to a supervised downstream task (Howard and Ruder, 2018; Devlin et al., 2018; Radford et al., 2018). Our work is more similar to the former, since we fine-tune a parent category model in order to obtain a model for its subcategory – transfer from supervised data.

3 Proposed Approach

We propose a minimalistic model architecture based on Gated Recurrent Unit (GRU) (Cho et al., 2014) combined with an attention (Dzmitry et al., 2015) mechanism. We use a bidirectional GRU to encode both forward and backward sequences. GRU can memoize the context of the text documents while the attention layer allows the model to selectively focus on important elements in the text. Our attention model closely follows the word attention module from (Yang et al., 2016).

Our model architecture is shown in Figure 1. The word sequences are fed into the GRU as embeddings. We use pre-trained embeddings from Glove (Pennington et al., 2017). Each state s_t produced by the GRU is a combination of s_{bt} and s_{ft}, where b and f denote the backward and forward hidden states, respectively, for each timestep t. As shown in the equations below, $\mathbf{S}$ denotes states for all the timesteps $(1, 2, ..., T)$. We apply attention on top of the GRU states to produce a fixed-dimensional vector representation Att(S). Furthermore, we combine a max-pooled (Maxpool) and mean-pooled (Meanpool) representation of all the GRU hidden states along with the $Att(S)$ vector to produce R – the sequence representation that is fed into the output layer.

$$\mathbf{S} = [s_1, s_2, s_3, ...s_T]$$
$$R = [Att(\mathbf{S}), Maxpool(\mathbf{S}), Meanpool(\mathbf{S})]$$

Finally, the output layer of the model includes a fully connected layer with sigmoid activations. The dimensionality of the fully-connected layer is determined by the number of categories in the classification task.

HTrans (Hierarchical Transfer Learning) is based on a recursive strategy of training parent and child category classifiers. Say, P1 is a top-level category with C1 as one of its children. Also, lets consider C12 as a child of C1. First, we train a binary classifier for P1. Documents in the training data that contain P1 as one of the labels are treated as positive instances, the rest are all negative. Next, we initialize the C1 binary classifier with the final model parameters of P1 classifier. After training the C1 classifier, the C12 classifier is initialized with parameters from C1 and so on. Following recent work on transfer learning in other domains (Hoo-Chang et al., 2016), we re-initialize the parameters of the final output layer randomly but retain the parameters of other layers.

Recent work on transfer learning (Howard and Ruder, 2018) suggested to use different learning rates for different layers. Based on recent findings in transfer learning (Bowman et al., 2015), we apply lower learning rates to the transferred parameters (from the parent classifier) and higher learning rates to the final fully connected classification (output) layer. We use Adam (Kingma and Ba, 2014) as our optimizer. We set the learning rate of the fully connected layer to 0.001 (high) as all the parameters in the layer are randomly initialized and they should be readjusted to the best possible values. In contrast, the learning rate for the other layers (GRU and attention) are changed to 0.0005 (low) to retain parent classification knowledge. In addition to different learning rates, we also freeze the embedding layer (Hu et al., 2014) after the top level classifiers have been trained. Layer freezing prevents over-fitting classifiers for categories in lower levels of the taxonomy.

4 Experimental Results

In this section, first, we describe the characteristics of the dataset followed by implementation details. Thereafter, we describe the experiments we conduct along with the results obtained.

Dataset: We use the Reuters dataset (RCV-v1) as provided in (Lewis et al., 2004). The dataset is a human-labeled collection of Reuters News articles from 1996-1997. There are a total of 103 categories according to the taxonomy. The dataset consists of 23,149 training and 784,446 testing documents, respectively.

Model	Micro-F1	Macro-F1
DGCNN	0.7618	0.4334
GRU-Att-basic	0.7980	0.5166
GRU-Att (class weights)	0.7974	0.5669
HTrans	**0.8051**†	**0.5849**†

Table 1: Comparison of Models on RCV1 dataset (†: Statistically significant at p≤0.05 compared to GRU-Att (with class weights))

Implementation and Metrics: We implemented our proposed network using PyTorch[1]. We use a 1 layer GRU with 96 hidden units and attention was added on top of the GRU layer. A dropout probability of 0.4 was applied on the GRU output. We use 100-dimensional pretrained word embeddings from Glove (Pennington et al., 2014). Each of the binary classifiers is trained for 10 epochs with early stopping (Caruana et al., 2001) with patience level 3. We use a batch size of 128 units for all our experiments. Models are trained on 2 Tesla V100 GPUs. The data corresponding to each category was randomly split into 85% training and 15% validation instances. We restrict the documents in the dataset to a maximum of 100 words from the body of the documents[2].

We use Binary Cross Entropy as the loss function for the classification problem. Due to significant data imbalance in several categories, we experiment with multiple class weights – 1, 2, 3, 5, 10, 30, 50 for each binary classifier and finally choose the best model based on validation metrics.

Metrics: We follow the most recent work (Peng et al., 2018) on RCV1 dataset and report Micro-F1 and Macro-F1 scores for our experiments. Micro-F1 considers the global precision and recall of the categories while Macro-F1 computes the average of the F1 scores obtained by individual categories.

4.1 Comparison of Different Models

We show the comparison of different approaches on the RCV1 dataset in table 1[3]. We refer to a version of GRU-Att without class weight optimization (default: 1) as GRU-Att-basic. As can be seen from the table, GRU-Att-basic performs significantly better than DGCNN on both Micro-

[1] https://pytorch.org/

[2] We tokenize using spacy: https://spacy.io/

[3] For comparisons with other models, please refer to (Peng et al., 2018)

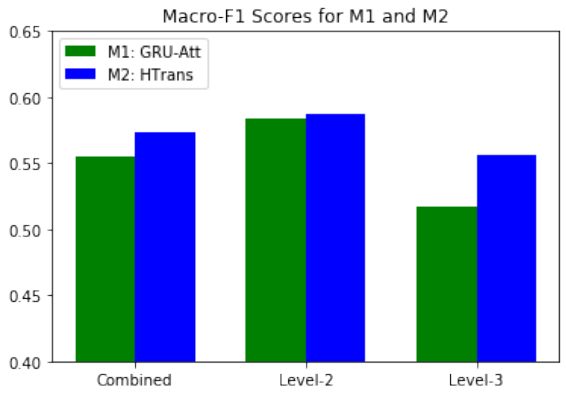

Figure 2: RCV1 dataset Levels 2 and 3: Macro-F1 without and with Transfer Learning

F1 (0.7980 vs 0.7618) and Macro-F1 (0.5166 vs 0.4334) scores, respectively. Using binary classifiers with a very basic architecture beats DGCNN easily.

Addition of class weights during model training (GRU-Att) further improves the binary models. We optimize the class weights based on the F1-score on the validation data. As can be seen from the table, Macro-F1 improves by close to 10% after incorporating class weights. The Micro-F1 remains unchanged, though. Therefore, the biggest benefit of using class weights is observed in categories where the number of instances during training is very low.

HTrans, our proposed technique that uses transfer learning (with embedding freezing and differential learning rates), further improves on GRU-Att by more than 3% on the Macro-F1 scores. Our initial conjecture was that transfer learning should help categories located at lower levels in the taxonomy. Therefore, we wanted to see the impact of HTrans on categories in different levels. Figure 2 shows the differences in Macro-F1 scores for the GRU-Att model (with class weights) and HTrans across different levels - Combined (level 2 and 3 both), level 2 and level 3. As can be seen from the Macro-F1 scores, HTrans outperforms GRU-Att at both levels – level 2 (0.587 vs 0.584) and 3 (0.556 vs 0.517). As expected, the improvement is visible in level 3 ($\sim$7%) with more clarity as level 3 contains the least number of training instances in the hierarchy.

Multi-label Model: We realize that training and inference using multiple binary classifiers might be a bottleneck due to resource constraints. In such cases, a single multi-label model might be preferred over multiple binary classifiers.

Model	Micro-F1	Macro-F1
DGCNN	0.7618	0.4334
GRU-Att-Multi (no weights)	0.7407	0.3937
GRU-Att-Multi (weights)	**0.7654**	**0.4842**

Table 2: Comparison of Multi-label Models on RCV1 dataset (weights imply the use of class weights during training)

To this end, we build a multi-label version of GRU-Att, GRU-Att-Multi, by replacing the output layer. Instead of a single output, it contains 103 output nodes (for the number of classes) for the RCV1 dataset. We wanted to investigate the use of class weights on the multi-label model. To select class weights on the multi-label model using a search over user-provided weights, we will have to evaluate an intractable number of class weight combinations. For example, say, we have two class weight options for each category. For 103 categories, it would result in trying out 2^{103} combinations of class weights making it impractical. Instead, we propose using the optimal class weights obtained from training the binary models and using them for the multi-label model training. We optimize the weighted F1-score during training the multi-label model. Loss function and optimizers are kept unchanged.

As can be seen from table 2, the use of the optimal class weights obtained from binary classifiers improve the Micro-F1 and Macro-F1 scores significantly on the multi-label model. The Macro-F1 scores suffer without the use of class weights. A more interesting observation is that our GRU-Att-Multi model trained using class weights outperforms the state-of-the-art multilabel model (DGCNN) on both metrics. The improvement of 12% seen in Macro-F1 score over DGCNN can be totally attributed to the class weighting scheme. We employ a much simpler architecture without the use of any regularization constraint but still can outperform DGCNN on both metrics.

5 Conclusions and Future Work

In this work, we propose HTrans, a hierarchical transfer learning-based strategy to train binary classifiers for categories in a taxonomy. Our approach relies on re-using model parameters trained at upper levels in the taxonomy and fine-tuning them for classifying categories at lower levels.

Our experiments on the RCV1 dataset show that classifiers of categories with less training examples benefit using pre-trained model parameters from upper level categories. Furthermore, we show that binary classifiers greatly outperform multi-label models. Finally, we show improvement over the state of the art multi-label model by using optimized class weights obtained when training the binary classifiers. As future work, we will investigate approaches to hyperparameter tuning to find better model architectures for hierarchical multi-label text classification tasks.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Jonathan Burez and Dirk Van den Poel. 2009. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636.

Rich Caruana, Steve Lawrence, and C Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bahdanau Dzmitry, Cho Kyunghyun, and B Yoshua. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Siddharth Gopal and Yiming Yang. 2013. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 257–265. ACM.

Yuhong Guo and Suicheng Gu. 2011. Multi-label classification using conditional dependency networks. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, page 1300.

Shin Hoo-Chang, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. 2016. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

Sheng-Jun Huang and Zhi-Hua Zhou. 2012. Multi-label learning by exploiting label correlations locally. In *Twenty-sixth AAAI conference on artificial intelligence*.

Diederik P Kingma and Jimmy Lei Ba. 2014. Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*.

Bartosz Krawczyk. 2016. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517. Association for Computational Linguistics.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, Austin, Texas. Association for Computational Linguistics.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1063–1072. International World Wide Web Conferences Steering Committee.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2017. Glove: Global vectors for word representation. 2014. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Grigorios Tsoumakas, Anastasios Dimou, Eleftherios Spyromitros, Vasileios Mezaris, Ioannis Kompatsiaris, and Ioannis Vlahavas. 2009. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st International Workshop on Learning from Multi-label Data*, pages 101–116.

Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Min-Ling Zhang and Kun Zhang. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 999–1008. ACM.

Denny Zhou, Lin Xiao, and Mingrui Wu. 2011. Hierarchical classification via orthogonal transfer.