Multi-Label Classification using Few-Shot Learning Techniques

Siddhant Garg
150711
siddhant@iitk.ac.in

Piyush Bagad
150487
bpiyush@iitk.ac.in

Priyadarshini Chintha
150206
priyach@iitk.ac.in

Bhavy Khatri
150186
bhavy@iitk.ac.in

Prakharji Gupta
150501
prakharg@iitk.ac.in

Rohan Kumar
150592
krohan@iitk.ac.in

Abstract

Multi-label classification is a long-studied problem in the machine learning research community. It finds numerous applications in real-world scenarios like gene function classification, semantic scene classification, text classification. Several classical approaches to the problem involve probabilistic modeling - which can also handle missing labels and explicitly model label correlations. Recent methods rely on deep learning techniques to learn latent embeddings of features as well as the labels. However, either of these methods require huge amounts of training data to be able to obtain a reliable model. Despite assuming availability of a large dataset, it is worth noting that we may have some labels which include classes that have very few examples labeled for those classes. Thus, for either of the aforementioned methods, the learnt model may fail to incorporate reliable knowledge of such *rarely occurring* classes during test time. We propose usage few-shot learning techniques to address the problem of such rarely occurring classes. The primary difference is that in the former, we assume that each of the classes have very few examples whereas in multi-label classification, there may be only certain classes of few labels that have one or few examples. We record our results using the naive approach that assumes independence among labels i.e. we solve a few-shot learning problem for each of the labels separately and incorporate all the models together during test time.

1 Introduction

The problem of multi-label classification has been of keen interest to the machine learning research community for a long time. It also is of interest to the industries leveraging large-scale machine learning and artificial intelligence for various tasks including text classification, semantic image annotation, query-keyword suggestion and various others in computational biology. The primary task in multi-label classification can be stated as follows: Given an input example $\mathbf{x} \in \mathbb{R}^D$, we need to predict a label vector $\mathbf{y} \in \{0, 1\}^L$, where D is the data dimensionality and L is the number of labels. We say that a label with value 1 is positive and the others as negative.

We note that in the problem of multi-label classification, we may have instances where in spite of having large number of examples globally, we may have one or more labels with a very few positive or negative examples. If a label has large number of positive examples, naturally it will have low number of negative examples. We call such classes for certain labels as *rarely occurring* classes. Deep learning methods may appear to work well globally but may be getting away by ignoring such classes. This may not be desirable since we may have test examples that correspond to such *rarely occurring* classes. The domain of few-shot classification involves learning from data which has very few examples from each of the classes. Note, that the problem of certain labels with one of the positive or negative classes being rare is different than the standard few-shot learning problem because in the latter we assume that each of the classes have one or few examples. We consider recent developments in this domain to address the problem of such rarely

Submitted as project report for the course CS771A.

occurring classes corresponding to some labels in multi-label classification setting. We begin with the naive approach that assumes label independence. In particular, we use prototypical networks - prototype based classification in a learnt embedding space independently for each of the labels. We also discuss some of the other ideas from the few-shot learning domain to solve the problem in consideration.

In section 3, we formally state the problem in detail. We follow it up with a fairly thorough literature survey of the domains: multi-label classification, few-shot learning and their intersection in section 2. We present a detailed report of our approach and the models used in section 3.1 and record the experimental settings and results in section 4. We conclude with a few ideas on potential future work in the domain in section 5.

2 Related Work

The problem in consideration lies in the intersection of the domains of multi-label classification and few-shot learning. The related work in corresponding areas has been sketched in detail.

2.1 Multi-label classification

Multi-label classification is a long studied problem in machine learning research. The key challenge in multi-label learning is handling the output space. The label sets grows exponentialy when the class labels increases. Several classical approaches to the problem involve probabilistic modeling - which can also handle missing labels and explicitly model label correlations. Based on the order of correlations the three types of strategies that are being considered for multi-label classification are: 1. *first order strategy*, where the problem is tackeled *label-by-label* and thus ignoring the dependencies of the other labels. Recent methods rely on deep learning techniques to learn latent embeddings of features as well as the labels. 2. *Second-order strategy* handles pairwise correlations among the labels. 3. *Higher order strategy* considers the influence of the other labels on a particular label. It is, however, computationaly more demanding and less scalable [15]. The multi-label algorithms are categorized as *Problem Transformation Methods* and *Algorithm Adaptation Methods*. Problem transformation methods tackles the multi-label problem by transforming it into well established learning scenarios. Binary Relevance, Classifier Chains, and Calibrated Label Ranking are some of the examples of this category. Whereas algorithms like ML-kNN (adapting lazy learning), ML-DT, Rank-SVM are examples of Algorithm Adaption Methods.

2.2 Few-shot learning

The problem of few-shot learning has gathered significant amount of attention in the recent years. Existing literature in this domain can be broadly classified into: the first approach involves learning an embedding with powerful discriminative features that can generalize well and the second approach relies on generative modeling generating examples of the classes with very few examples ([5, 10, 8]). From the former domain, *Matching networks* [14] tries to minimize the one shot classification error by matching the conditions of training to those of testing with the motive of achieving good generalization. *Prototypical networks* based approach [12] achieves compelling results using simple prototype based classification method in an embedded feature space. Several other works [1, 2, 4] use various techniques to embed the data in a new feature space and use the obtained feature space to achieve the goal of one or few-shot learning.

2.3 Zero and Few-shot learning for multi-label classification

The core idea this project is based upon is to leverage few-shot learning methods to improve multi-label classification performance in cases where some of the labels may have rarely occurring classes. This idea has been explored in the literature but it is fair to say that it is in its nascent stage. Zero-shot learning differs from few-shot learning in the sense that it has no examples examples whatsoever of some of the classes but additionally has some sort of meta-data about those classes. It may still be of interest to look at techniques that extend ZSL to multi-label scenario. Sve et. al [13] introduce ways of extending the usual zero-shot learning scenario to multi-label classification. [7] use deep learning to address multi-label zero shot learning by incorporating knowledge graphs to exploit relationship among labels. [11] leverages ideas from matching networks [14] for text classification in the medical domain. [6] is the closest work to this project which incorporates prototypical networks with training methods borrowed from matching networks for multi-label genomic classification.

2.3.1 Prototypical Networks: An overview

Prototypical networks are used for few-shot classification task. The classifier learns a metric space in which we project the a few number of examples of each class, not seen in the training, and learn protypical representations of each class. Classification is then performed by computing the distances from the prototypes in the learnt metric space. The key step involved in learning the prototypical network is using the episodic training, that was used in *Matching Nets*. We followed the same approach and created support sets and query set for each episode of the training, the details of which is presented in section 3.1.

3 Problem Statement

Multi-Label Classification: We will follow the definition given in the review of Multi-Label learning algorithms[15]. Suppose $\mathcal{X} = \mathbb{R}^D$, denotes a D-dimensional vector space, and $\mathcal{L} = \{k^1, \dots, k^\ell\}$ denotes the label space with ℓ possible class labels. The multi-label classification task is to learn a classifier $h : \mathcal{X} \rightarrow 2^{\mathcal{L}}$ from the multi-label training data $\mathcal{D} = \{(\mathbf{x}_i, Y_i) | 1 \leq i \leq N\}$, where $\mathbf{x}_i \in \mathcal{X}$ and $Y_i \subseteq \mathcal{L}$ as the set of associated labels with $\mathbf{x}_i$.

We tackeled the problem of multi-label classification in the few-shot scenario. The final model will generalizes to new classes not seen in the training using very few examples from each class and predict the label set of for the query samples.

3.1 Our Approach

We used the first-order approach for this task where we treated all the ℓ labels independently and learn ℓ different non-linear mappings $\{f_{\phi_i}\}_{i=1}^{\ell}$ for each label. Here $f_{\phi_i} : \mathbb{R}^D \rightarrow \mathbb{R}^M$ denotes feature embedding in $\mathbb{R}^M$ space corresponding to each label k^i with learnable parameters ϕ_i. After that we use prototype based classification on the embedded space. Our training algorithm mimics the algorithm given in the prototypical learning paper [12].

Also, for the label k^j, let S^j denotes the set of examples: $S^j = \{(\mathbf{x}_1, y_1^{(j)}), \dots, (\mathbf{x}_N, y_{N_S}^{(j)})\}$, N_S is the number of examples in support set. Let $\mathbf{c}^j_+$, $\mathbf{c}^j_- \in \mathbb{R}^{M_j}$ denote the prototype of each class of ith label. The prototypes are the mean vectors of the embedded +ve (belonging) and -ve (non-belonging) points in the training data. [12]

$$\mathbf{c}^j_+ = \frac{1}{|S^j|} \sum_{\mathbf{x}_i \in S^j} f^j_{\phi_j}(\mathbf{x}_i) \quad \mathbf{c}^j_- = \frac{1}{N - |S^j|} \sum_{\mathbf{x}_i \notin S^j} f^j_{\phi_j}(\mathbf{x}_i)$$

Now if we take a distance function $d : \mathbb{R}^{M_j} \times \mathbb{R}^{M_j} \rightarrow [0, \infty)$ then prototypical network produces a distribution over classes for a query point $\mathbf{x}_*$ based on a softmax over distances to the prototypes in the embedding space [12]. The probability of a point belonging to a particular class depends on the distance of the point from prototypical mean of that class. Closer the mean from point imply that there is higher probability that point will belong to that class. This can be captured by the following equation:

$$p_{\phi_i}(k^j \in Y_* | \mathbf{x}_*) = \frac{exp(-d(f^j_{\phi_j}(\mathbf{x}_*), \mathbf{c}^j_+))}{exp(-d(f^j_{\phi_j}(\mathbf{x}_*), \mathbf{c}^j_+)) + exp(-d(f^j_{\phi_j}(\mathbf{x}_*), \mathbf{c}^j_-))}$$

We will try to minimize negative log probability $J^{(i)}(\phi) = -\log p_{\phi_i}(y^i = k)$ of the true class (+ve/-ve). We trained the model for each label using adam optimizer in Pytorch[9]. Predictions were made for each label separately.

Algorithm 1 Algorithm for training episodic loss computation. N is the number of examples in the dataset. N_S is the number of examples in the support set per class. N_Q is the number of query examples per class. The number of classes is 2 (one for positive class and one for negative class). ℓ is the size of the label set. Let $randomsample(A, M)$ be the set of M random points from set A.

```
for $i = 1, \ldots, \ell$ do
    Let $\mathcal{D} = \{(\mathbf{x}_1, y_1^{(i)}), \ldots, (\mathbf{x}_N, y_N^{(i)})\}$ be the training dataset, where we are considering the $i^{th}$ label in the label set. $y^{(i)} \in \{1, -1\}$.
    $\mathcal{D}_+ = \{(\mathbf{x}_j, y_j^{(i)}) | y_j^{(i)} = 1, j = 1, \ldots, N\}$ and $\mathcal{D}_- = \{(\mathbf{x}_j, y_j^{(i)}) | y_j^{(i)} = -1, j = 1, \ldots, N\}$.
    $S_+ = randomsample(\mathcal{D}_+, N_S)$ and $S_- = randomsample(\mathcal{D}_-$
    $Q_+ = randomsample(\mathcal{D}_+ \setminus S_+, N_Q)$ and $Q_- = randomsample(\mathcal{D}_- \setminus S_-, N_Q)$.
    $\mathbf{c}_+ = \frac{1}{N_S} \sum_{(\mathbf{x}_j, y_j^{(i)}) \in S_+} f_{\phi_i}(\mathbf{x}_j)$ and $\mathbf{c}_- = \frac{1}{N_S} \sum_{(\mathbf{x}_j, y_j^{(i)}) \in S_-} f_{\phi_i}(\mathbf{x}_j)$.
    $J^{(i)} \leftarrow 0$.
    for $k \in \{+, -\}$ do
        for $(\mathbf{x}, y) \in Q_k$ do
            $J^{(i)} \leftarrow J^{(i)} + \frac{1}{2N_Q} \left[ d(f_{\phi_i}(\mathbf{x}), \mathbf{c}_k) + \log \sum_{k'} \exp(-d(_{\phi_i}(\mathbf{x}), \mathbf{c}_{k'})) \right]$
```

4 Experiments and Results

Dataset: For few-shot multi-label classification task, we performed our experiments on the *Multi-Instance Multi-Label Learning* dataset [3]. The image data set consists of 2,000 natural scene images. The label set consists of 5 classes namely, desert, mountains, sea, sunset and trees. The number of images belonging to more than one class (e.g. sea+sunset) comprises over 22% of the data set, many combined classes (e.g. mountains+sunset +trees) are extremely rare. On average, each image is associated with 1.24 class labels. Each image is 9×15 dimensional feature matrix. We padded the images with zeros evenly to make them 28×28 dimensional matrices.

Architecture: Our embedding architechture is same as used in the prototypical networks[12] with additional regularization by applying dropout after the convolutional layer. Our model consists of four convolutional blocks. Each block comprises a 64-filter 3×3 convolution, batch normalization layer and dropout with probability of an element being zeroed is 0.2, a ReLU nonlinearity and a 2×2 max-pooling layer. When applied to the 28×28 images, this architecture results in a 64-dimensional output space. We use the same encoder for embedding both support and query points. All of our models were trained via SGD with Adam.We used the learning rate of 10^{-3}. The complete code was written in pytorch from scratch.

Training and Testing: We created ℓ different datasets to train ℓ different models. Each dataset consists of same observation points $\mathbf{x}_i$, but different labels. For the i^{th} datset, $\{(\mathbf{x}_1^{(i)}, y_1^{(i)}), \ldots, (\mathbf{x}_N^{(i)}, y_N^{(i)})\}$, $i \in \{1, \ldots, \ell\}$ label $y_j^{(i)} = 1$ if $y_j^{(i)} \in Y_j$ in the original dataset, and $y_j^{(i)} = 0$, otherwise, $j \in \{1, \ldots, N\}$. We trained our model with the Euclidean Distance. It was shown to be the best distance metric in case of prototypical learning[12]. We trained our model with 10, 500, 1000 randomly generated episodes. For the i^{th} model, the support and query set contains 10 examples having the $y_j^{(i)} = 0$ and 10 examples with the $y_j^{(i)} = 0$.

Evaluation Metrics: The following evaluation metrics are used for the final results.

- Accuracy

$$A = \frac{1}{N} \sum_{n=1}^{N} \frac{|Y_n \cap \hat{Y_n}|}{|Y_n \cup \hat{Y_n}|}$$

- Precision

$$P = \frac{1}{N} \sum_{n=1}^{N} \frac{|Y_n \cap \hat{Y_n}|}{|\hat{Y_n}|}$$

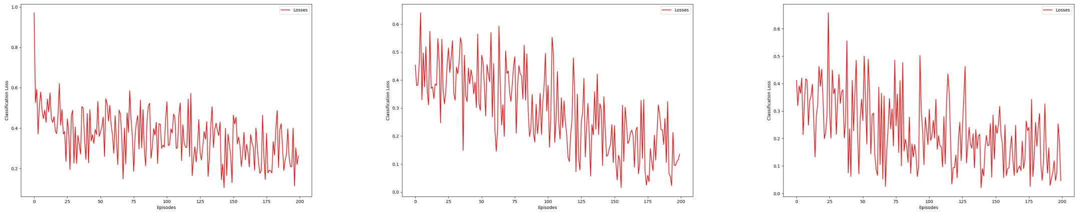
- Recall

$$R = \frac{1}{N}\sum_{n=1}^{N}\frac{|Y_n \cap \hat{Y_n}|}{|Y_n|}$$

Results: The following table presents the results of our experiment with the model on the dataset.

Number of episodes	Accuracy	Precision	Recall
10	0.17	0.18	0.34
500	0.22	0.21	0.45
1000	0.24	0.22	0.48

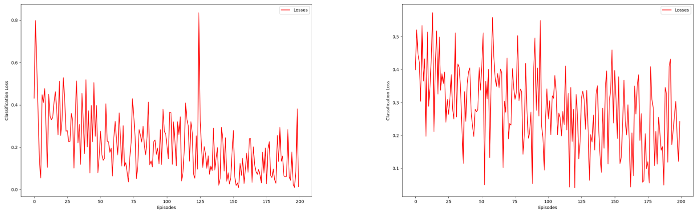
Below are the graphs of loss values vs episodes for each label, on the query set. The graph is eratic but the general trend is decreasing.



(a) Label 1 Loss

(b) Label 2 Loss

(c) Label 3 Loss



(a) Label 4 Loss

(b) Label 5 Loss

5 Discussion and Future Work

5.1 Regularizer based on `word2vec` training

The major problem with naive approach is that it will not capture the co-relation between various labels. For e.g. we have the two class labels as gender and occupation s.t.:

Gender	Occupation
Male	Plumber
Female	Receptionist

Table 1: Example illustrating importance of label correlations.

Now, it may be possible that certain gender may have higher co-relation with certain occupation. But, our naive approach will not be able to capture this co-relation. So one of our goals will be to "think of ways to incorporate these kind of co-relations in our model". Formally, instead of modeling $p(y_i|\mathbf{x})$, we would want to model $p(y_i|\mathbf{x}, \{y_j\}_{j\neq i})$. One of the immidiete future works would be to consider a word-embedding like training procedure in which we regularize the model to force the prototypes of those classes whose labels have high correlation to be closer in the embedding space.

5.2 Extending Siamese networks for multi-label classification

Siamese network has been used for the task of one-shot image classification in the multi class setting [4]. Deep learning model need a lot of data to be trained reasonably well. In Koch et. al authors exploited the power of neural networks for the task of one and few shot learning using "twin" network - namely *siamese network.* Instead of taking single feature vector as an input, a pair of examples was trained on two identically symmetric network which was then joined by the energy function at the top. This function computes some metric between the highest level feature on each side. The output of the network predicted the similarity between two feature vector. To specify the label for new instance, similarity was calculated for each instance in the support class against the test instance. [4] Label assigned for the test instance was that of the opposite class for which the pairwise score was highest. Note that, Pairings which are most similar to each other was awarded the highest score. The most natural advantage of using this approach is that even if we originally have $\mathcal{O}(n)$ data points, the effective dataset will be of order $\mathcal{O}(n^2)$.Please see Figure 3 of the siamese network.

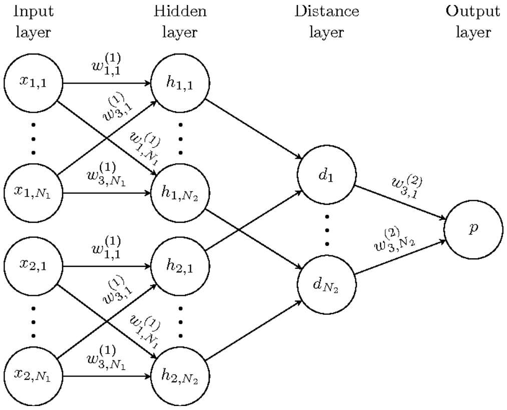


Figure 3: A simple 2 hidden layer siamese network for binary classification with logistic prediction *p*. The structure of the network is replicated across the top and bottom sections to form twin networks, with shared weight matrices at each layer. [4]

Suppose the total number of label is *L*. We propose to use *L* nodes in the output layer one for each label. This will be followed by sigmoid activation function at each output node. The rest of the architecture will remain the same. Suppose for the feature input pair $\mathbf{x}_{12} = (\mathbf{x}_1, \mathbf{x}_2)$ the corresponding output label vectors are $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively, where $\mathbf{y}_1, \mathbf{y}_2$ represent one hot encoding labels. Then our similarity vector $\mathbf{y}_{12} = \{XOR(\mathbf{y}_1, \mathbf{y}_2)\}^C$ represent the similarity output for our *siamese network.* Here $XOR(.,.)$, $(.)^C$ represent the component wise *xor* and *complement* operations respectively. The prediction on the test data will be pretty much similar to what we did in the single label case. Please see Figure 4 of the proposed network.

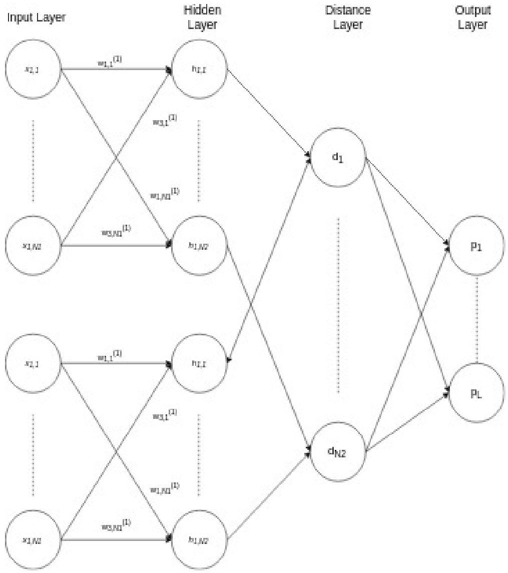


Figure 4: Proposal network for multi-label classification

References

[1] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 523–531. Curran Associates, Inc., 2016.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[3] Zhi hua Zhou and Min ling Zhang. Multi-instance multi-label learning with application to scene classification. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1609–1616. MIT Press, 2007.

[4] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

[5] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[6] Jack Lanchantin, Arshdeep Sekhon, Ritambhara Singh, and Yanjun Qi. Prototype matching networks for large-scale multi-label genomic sequence classification. *CoRR*, abs/1710.11238, 2017.

[7] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. *CoRR*, abs/1711.06526, 2017.

[8] Akshay Mehrotra and Ambedkar Dukkipati. Generative adversarial residual pairwise networks for one shot learning. *CoRR*, abs/1703.08033, 2017.

[9] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[10] Danilo Rezende, Shakir, Ivo Danihelka, Karol Gregor, and Daan Wierstra. One-shot generalization in deep generative models. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1521–1529, New York, New York, USA, 20–22 Jun 2016. PMLR.

[11] Anthony Rios and Ramakanth Kavuluru. EMR coding with semi-parametric multi-head matching networks. In *NAACL-HLT*, pages 2081–2091. Association for Computational Linguistics, 2018.

[12] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4077–4087. Curran Associates, Inc., 2017.

[13] Bjørnar Moe Sve, Thomas; Remmen. Investigating zero-shot learning techniques in multi-label scenarios. -, 2017.

[14] Oriol Vinyals, Charles Blundell, Tim Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016.

[15] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2014.

Disclaimer

We hereby declare that the work presented in the project report entitled "Multi-Label Classification using Few-Shot Learning Techniques" contains our own ideas in our own words. At places, where ideas and words are borrowed from other sources, proper references, as applicable, have been cited. To the best of our knowledge this work does not emanate or resemble to other work created by person(s) other than mentioned herein.