Initializing neural networks for hierarchical multi-label text classification

Simon Baker [1,2] **Anna Korhonen** [2]
[1]Computer Laboratory, 15 JJ Thomson Avenue
[2] Language Technology Lab, DTAL
University of Cambridge, UK
simon.baker@cl.cam.ac.uk, alk23@cam.ac.uk

Abstract

Many tasks in the biomedical domain require the assignment of one or more predefined labels to input text, where the labels are a part of a hierarchical structure (such as a taxonomy). The conventional approach is to use a one-*vs*.-rest (OVR) classification setup, where a binary classifier is trained for each label in the taxonomy or ontology where all instances not belonging to the class are considered negative examples. The main drawbacks to this approach are that dependencies between classes are not leveraged in the training and classification process, and the additional computational cost of training parallel classifiers. In this paper, we apply a new method for hierarchical multi-label text classification that initializes a neural network model final hidden layer such that it leverages label co-occurrence relations such as hypernymy. This approach elegantly lends itself to hierarchical classification. We evaluated this approach using two hierarchical multi-label text classification tasks in the biomedical domain using both sentence- and document-level classification. Our evaluation shows promising results for this approach.

1 Introduction

Many tasks in biomedical natural language processing require the assignment of one or more labels to input text, where there exists some structure (such as a taxonomy or ontology) between the labels: for example, the assignment of Medical Subject Headings (MeSH) to PubMed abstracts (Lipscomb, 2000).

A typical approach to classifying multi-label documents is to construct a binary classifier for each label in the taxonomy or ontology where all documents not belonging to the class are considered negative examples, *i.e.* one-*vs*.-rest (OVR) classification (Hong and Cho, 2008). This approach has two major drawbacks: first, it makes the hard assumption that the classes are independent which often does not reflect reality; second, it is more computationally expensive (albeit by a constant factor): if there are a very large number of classes, the approach becomes computationally unrealistic.

In this paper, we investigate a simple and computationally fast approach for multi-label classification with a focus on labels that share a structure, such as a hierarchy (taxonomy). This approach can work with established neural network architectures such as a convolutional neural network (CNN) by simply initializing the final output layer to leverage the co-occurrences between the labels in the training data.

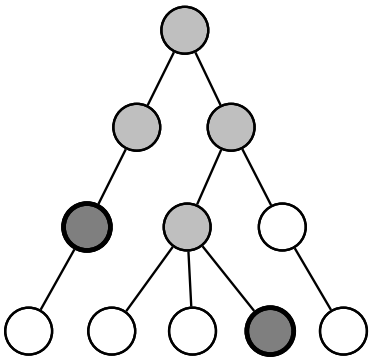

Figure 1: Hierarchical multi-label classification. Nodes represent possible labels that can be assigned to text: a dark grey node denotes an explicit label assignment and light grey denotes implicit assignment due to a hypernymy relationship with the explicitly assigned label.

First, we need to define hierarchical multi-label classification. In multi-label text classification, input text can be associated with multiple labels (label co-occurrence). When the labels form a hierarchy, they share a hypernym–hyponym relation (Figure 1). When multiple labels are assigned to the text, if it is explicitly labeled by a subclass it must also implicitly include all of the its superclasses.

The co-occurrence between subclasses and superclasses as labels for the input text contains information we would like to leverage to improve multi-label classification using a neural network.

In this paper we experiment with this approach using two hierarchical multi-label text classification tasks in the biomedical domain, using both document- and sentence-level classification.

We first briefly summarize related literature on the topic of multi-label classification using neural networks, we then describe our methodology and evaluation procedure, and then we present and discuss our results.

2 Related work

There have been numerous works that focus on solving hierarchical text classification. Sun and Lim (2001) proposed top-down level-based SVM classification. More recently, Sokolov and Ben-Hur (2010); Sokolov et al. (2013) predict ontology terms by explicitly modeling the structure hierarchy using kernel methods for structured output space. Clark and Radivojac (2013) use a Bayesian network, structured according to the underlying ontology to model the prior probability.

Within the context of neural networks, Kurata et al. (2016) propose a scheme for initializing neural networks hidden output layers by taking into account multi-label co-occurrence. Their method treats some of the neurons in the final hidden layer as dedicated neurons for each pattern of label co-occurrence. These dedicated neurons are initialized to connect to the corresponding co-occurring labels with stronger weights than to others. They evaluated their approach on the *RCV1-v2* dataset (Lewis et al., 2004) from the general domain, containing only flat labels. Their evaluation shows promising results. However, their applicability to the biomedical domain with more a complex set of labels that share a hierarchy remains an open question.

Chen et al. (2017) propose a convolutional neural network (CNN) and recurrent neural network (RNN) ensemble method that is capable of efficiently representing text features and modeling high-order label correlation (including co-occurrence). However, they show that their method is susceptible to overfitting with small datasets.

Cerri et al. (2014) propose a method for hierarchical multi-label text classification that incrementally trains a multi-layer perceptron for each level of the classification hierarchy. Predictions made by a neural network in a given level are used as inputs to the neural network responsible for the prediction in the next level. Their method was evaluated against several datasets with convincing results.

There are also several relevant works that propose the inclusion of multi-label co-occurrence into loss functions such as pairwise ranking loss (Zhang and Zhou, 2006) and more recent work by Nam et al. (2014), who report that binary cross-entropy can outperform the pairwise ranking loss by leveraging rectified linear units (ReLUs) for nonlinearity.

3 Method

In this section, we describe the approach of initializing a neural network for multi-label classification. We base our CNN architecture on the model of Kim (2014), which has been used widely in text classification tasks, but this approach can be applied to any other architecture.

Briefly, this model consists of an initial embedding layer that maps input texts into matrices, followed by convolutions of different filter sizes and 1-max pooling, and finally a fully connected layer. The architecture is illustrated in Figure 2.

To perform multi-label classification using this architecture, the final output layer uses logistic (sigmoid) activation function σ:

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{1}$$

where x is the input signal. The output range of the function is between zero and one; if it is above a cut-off threshold T_σ (which is tuned by grid search on the development dataset) then the prediction y'_k for label y_k is positive. We use a binary cross-entropy loss function L:

$$L(\theta, (x, y)) = -\sum_{k=1}^{K} y_k \log(y'_k) + (1-y_k) \log(1-y'_k) \quad (2)$$

where θ is the model parameters and K is the number of classes.

As shown in Figure 2, the multi-label initialization happens in output layer of the network. Figure 3 illustrates the initialization process. The rows represent the units in the final hidden layer, while the columns represent the output classes.

The idea is to initialize the final hidden layer with rows that map to co-occurrence of labels in the training data. This can be implicit hypernymy relations between the labels, or explicit co-occurrence in the annotation. For each co-occurrence, the value ω is assigned to the associated classes and a value of zero is assigned to the rest. The value ω is the upper bound of the normalized initialization proposed by Glorot and Bengio (2010), which is calculated as follows:

$$\omega = \frac{\sqrt{6}}{\sqrt{n_h + n_k}} \quad (3)$$

where n_h is the number of units in the final hidden layer and n_k is the number of units in the output layer (*i.e.* classes). This value was also successfully used by Kurata et al. (2016) in their initialization procedure.

The motivation for this initialization is to incline units in the hidden layer to be dedicated to representing co-occurrence of labels by triggering only the corresponding label nodes in the output layer when they are active.

The number of units in the final hidden layer can exceed the number of label co-occurrences in the training data. We must therefore decide what to do with the remaining hidden units. Kurata et al. (2016) assign random values to these units (shown in Figure 3 (B)). We will also use this scheme, but in addition we propose another variant: we assign the value zero for these neurons, so that the hidden layer will only be initialized with nodes that represent label co-occurrence.

We implement the neural network and the initialization using Keras (Chollet, 2015). the hyperparameters for our model and baselines are those of Kim (2014), summarized in Table 1.

We use word2vec embeddings trained on PubMed by Chiu et al. (2016).

Hyperparameter	Value
Word vector size	300
Filter sizes	3, 4, and 5
Number of filters	300 (100 of each size)
Dropout probability	0.5
Minibatch size	50
Input size (in tokens)	500 (documents), 100 (sentences)

Table 1: Our baseline model, based on Kim (2014) model hyperparameters.

4 Data

We investigate our approach using two multi-label classification tasks. In this section, we describe the nature of these tasks and the annotated gold standard data.

Task 1: The Hallmarks of Cancer The Hallmarks of Cancer describe a set of interrelated biological properties and behaviors that enable cancer to thrive in the body. Introduced in the seminal paper by Hanahan and Weinberg (2000)—the most cited paper in the journal *Cell*—the hallmarks of cancer have seen widespread use in BioNLP for many systems and works, including the BioNLP Shared Task 2013, 'Cancer Genetics task' (Pyysalo et al., 2013), which involved the extraction of events (*i.e.* biological processes) from cancer-domain texts. Baker et al. (2016) have released an expert-annotated dataset for cancer hallmark classification for both sentences and documents from PubMed. The data consists of multi-labelled documents and sentences using a taxonomy of 37 classes.

Task 2: The exposure taxonomy Larsson et al. (2017) introduce a new task and an associated annotated dataset for the classification of text (documents or sentences) for chemical risk assessment: more specifically, the assessment of exposure routes (such as ingestion, inhalation, or dermal absorption) and human biomonitoring (the measurement of exposure biomarkers). The taxonomy of 32 classes is divided into two branches: Biomonitoring and Exposure routes.

We split both datasets (by documents) into train, development (dev), and test splits in order to evaluate our methodology. Table 4 summarizes key statistics for these splits.

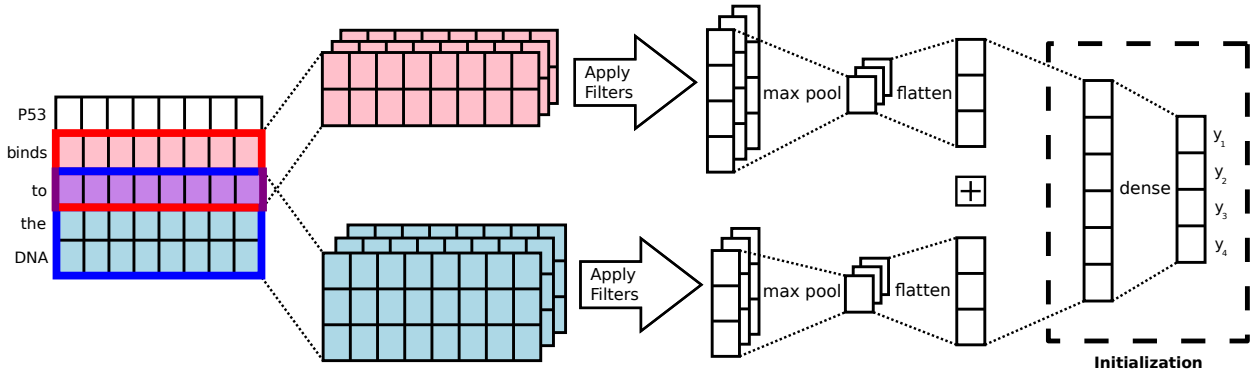

Figure 2: Convolutional Neural Network (CNN) architecture with the initialization layer outlined. The CNN architecture is based on the model of Kim (2014).

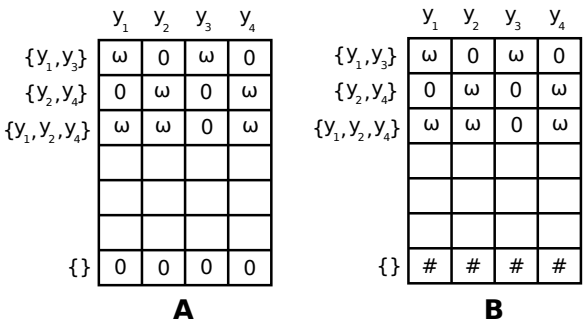

Figure 3: The two initialization schemes: (A) initializing non label co-occurrence nodes with zero, (B) initializing non label co-occurrence with a random value (#) drawn from a uniform distribution.

	Task 1		Task 2	
	Document	Sentence	Document	Sentence
Train	1,303	12,279	2,555	25,307
Dev	183	1,775	384	3,770
Test	366	3,410	722	7,100
Total	1,852	17,464	3,661	36,177

Table 2: Summary statistics for Tasks 1 and 2.

We also measure the overlap in the data between pairs of labels. We use Jaccard similarity J to measure this overlap using the following equation:

$$J(A, B) = \frac{A \cap B}{A \cup B} \quad (4)$$

Where A and B are sets of instances labelled with these classes. Table 4 summarizes the average and maximum pairwise Jaccard similarity between the labels in both tasks.

Table 4 shows that Task 1 labels have slightly more overlap than those of Task 2.

	Task 1		Task 2	
	Document	Sentence	Document	Sentence
Avg	4.1	2.3	5.7	3.0
Max	49.3	49.4	45.7	42.5

Table 3: Jaccard similarity scores (expressed as percentages) between label pairs.

The large difference in values between document and sentence label overlap is due to the fact that documents have more labels per instance than sentences. The average score is much lower as most pair combinations would not have overlaps; where there is overlap it is typically significant (as shown by the Max row in Table 4).

5 Evaluation

In this section, we describe our experimental setup and our baselines.

5.1 Experimental setup

We ascertain the performance of our approach under a controlled experimental setup. We compare two baseline models (described in the next section), and two variants of the initialization models corresponding to the two initialization schemes described in Figure 3. We will refer to the first scheme (allocating all units in the final hidden layer to representing label co-occurrences and zeroing all other units) as INIT-A, and the second scheme (allocating a random value drawn from a uniform distribution for non co-occurrence hidden units) as INIT-B. We use the hyperparameters in Table 1 and data splits in Table 4 for all models.

We check the model's performance (F_1-score) on development data at the end of every epoch. We

select the model from the best-performing epoch and train it until its performance does not improve for ten epochs.

5.2 Baselines

We compare two baselines in our setup: one-*vs*.-rest (OVR) and multi-label baseline (MULTI-BASIC)

One-*vs*.-rest (OVR) We train and evaluate K independent binary CNN classifiers (*i.e.* a single classifier per class with the instances of that class as positive samples and all other instances as negatives).

Multi-label baseline (MULTI-BASIC) We train and evaluate a multi-label baseline based on Figure 2 without initialization of the final hidden layer. This enables us to directly compare the effect of the initialization step. As with the initialization models (INIT-A and INIT-B), we grid search the sigmoid cut-off parameter T_σ on the development data at the end of each epoch to be used with the selected best model on the test split.

5.3 Post-processing label correction

The predicted output labels from all of our models can be inconsistent with respect to the label hierarchy: a subclass label might be positive while its superclass is negative, thereby contradicting the hypernmy relation (illustrated in Figure 4 (A)).

We can apply two kinds of post-processing corrections to the predicted labels in order for them to be well-formed. We call the first *transitive correction* (Figure 4 (B)), wherein we correct all superclass labels (transitively) to be positive. The alternative is *retractive correction* (Figure 4 (C)), where we ignore the positive classification of the subclass label, and accept only the chain of superclass labels (from the root), as long as they are well-formed.

We apply both of these post-processing correction policies to all of the models, and observe the effect on their performance.

6 Results

In this section, we describe the results for the evaluation setup described in the last section. We assess the performance of the models by measuring the precision (P), recall (R), and F_1-scores of the labels in the model using the one-*vs*.-rest setup.

	Document			Sentence		
	P	R	F_1	P	R	F_1
Task 1						
OVR	77.8	51.7	62.1	56.8	30.7	39.9
MULTI-BASIC	71.0	71.6	71.3	42.0	71.9	53.0
INIT-A	73.4	76.9	75.1	42.7	70.6	53.2
INIT-B	68.3	83.4	75.1	40.1	72.2	51.6
Task 2						
OVR	89.5	87.1	88.3	66.2	62.8	64.5
MULTI-BASIC	86.0	90.0	88.0	51.7	75.6	61.4
INIT-A	86.7	91.1	88.9	49.5	80.7	61.4
INIT-B	75.7	91.3	82.8	47.0	83.2	60.1

Table 4: Performance results for Tasks 1 and 2. All figures are micro-averages expressed as percentages.

Table 6 shows the micro-averaged scores across all labels for both tasks.

The results show that for Task 1, all multi-labeled models significantly outperform the OVR model in F_1-score, which is explained by a very substantial improvement in recall. INIT-A outperforms all models in this task, particularly at the document level where there is 5 point improvement over MULTI-BASIC.

The results for Task 2 on are more mixed. Overall, all models achieve a similar F_1-score at the document level. However, there is a clear improvement in recall at the cost of lower precision when compared to OVR. The best performing model at the document level is INIT-A. On the sentence level, OVR seems to outperform all multi-label models by a good margin. This indicates that the multi-label approach did not aid sentence-level classification in this particular task.

The figures in Table 6 do not show a complete picture as the interactions between the labels are not taken into account.

We can observe the proportion of the number of labels assigned to each instance by the classifiers, and compare these proportions to the annotated gold standard test data. Figure 5 shows this distribution for each classifier. We can see in Figure 5 that the overall distributions for all sentence-level classifiers (for both tasks) are closer to the gold standard distribution (compared to document level). This is due to the fact that most sentences have no assigned labels. For Task 2, the classifiers tend to assign more labels than the gold standard.

Document-level classification shows two out-

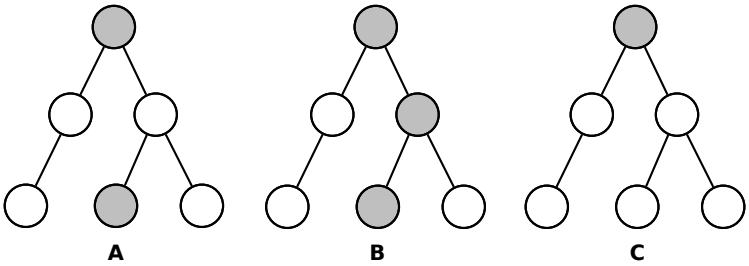

Figure 4: Illustrating post-processing label correction, with (A) showing the output prediction from the neural network model, (B) applying transitive correction, (C) applying retractive correction.

liers. For Task 1, we observe that OVR disproportionately assigns exactly one label per document compared to gold standard (where documents have two to three labels on average). In Task 2, INIT-B assigns more labels per document than the gold standard (and every other model).

In addition to looking at the number of labels per class, we also measure the proportion of exact label matches that each model predicts as shown in Table 6.

	Task 1		Task 2	
	Doc.	Sent.	Doc.	Sent.
OVR	18.0	67.9	43.4	61.7
MULTI-BASIC	26.2	59.3	40.9	54.2
INIT-A	33.9	65.9	45.6	53.1
INIT-B	31.3	62.6	12.7	49.7

Table 5: The proportion (%) of exact matches.

For document classification in Task 1, INIT-A outperforms all models, while OVR significantly underperforms. However, OVR performs significantly better than all other models when classifying sentences when considering exact matches only.

Finally, we look at how consistent (well-formed) the predictions output by each model are. We do this by running the post-processing label correction policies described in Section 5.3. Table 6 summarizes these results.

For Task 1, OVR shows the largest variance after the application of any method of correction, whereas no multi-labeled model shows this variation. This indicates that the post-processing corrections had little effect on the predicted results as they were already well-formed. For Task 2, there is very little variance for all multi-labeled models, with only a slight change for OVR.

	Document			Sentence		
	O	T	R	O	T	R
Task 1						
OVR	62.1	63.9	60.6	39.9	42.2	37.5
MULTI-BASIC	71.3	71.3	71.2	53.0	53.0	53.0
INIT-A	75.1	75.0	75.2	53.2	53.2	53.3
INIT-B	75.1	74.9	75.3	51.6	51.5	51.6
Task 2						
OVR	88.3	88.4	88.2	64.5	65.3	63.3
MULTI-BASIC	88.0	87.7	88.1	61.4	61.3	61.7
INIT-A	88.9	88.7	89.0	61.4	61.3	61.5
INIT-B	82.8	82.8	82.8	60.1	59.8	60.4

Table 6: Post-processing label correction. O is the predicted output, T is transitive correction, and R is retractive correction. All figures are micro-averaged F_1-scores expressed as percentages.

7 Discussion

The strength of using the hidden-layer initialization for multi-label classification lies in leveraging the co-occurrence between labels. Naturally, if such co-occurrences are relatively rare in the dataset, then this approach becomes less effective. This implies that this approach is especially attractive for hierarchical multi-label classification, because of the implicit hypernym–hyponym relations between the labels, which by definition guarantees co-occurrence of labels in the datasets. The superclass labels must be included when labeling a given example in order to model the hierarchical nature of the labels.

Another key strength of this approach is its low computational cost, which is only proportional to the size of the input text, and the number of label co-occurrences.

However, when there is a large amount of training data, the number of label co-occurrences can be larger than the number of the hidden units. In such a case, one possible option is to select an ap-

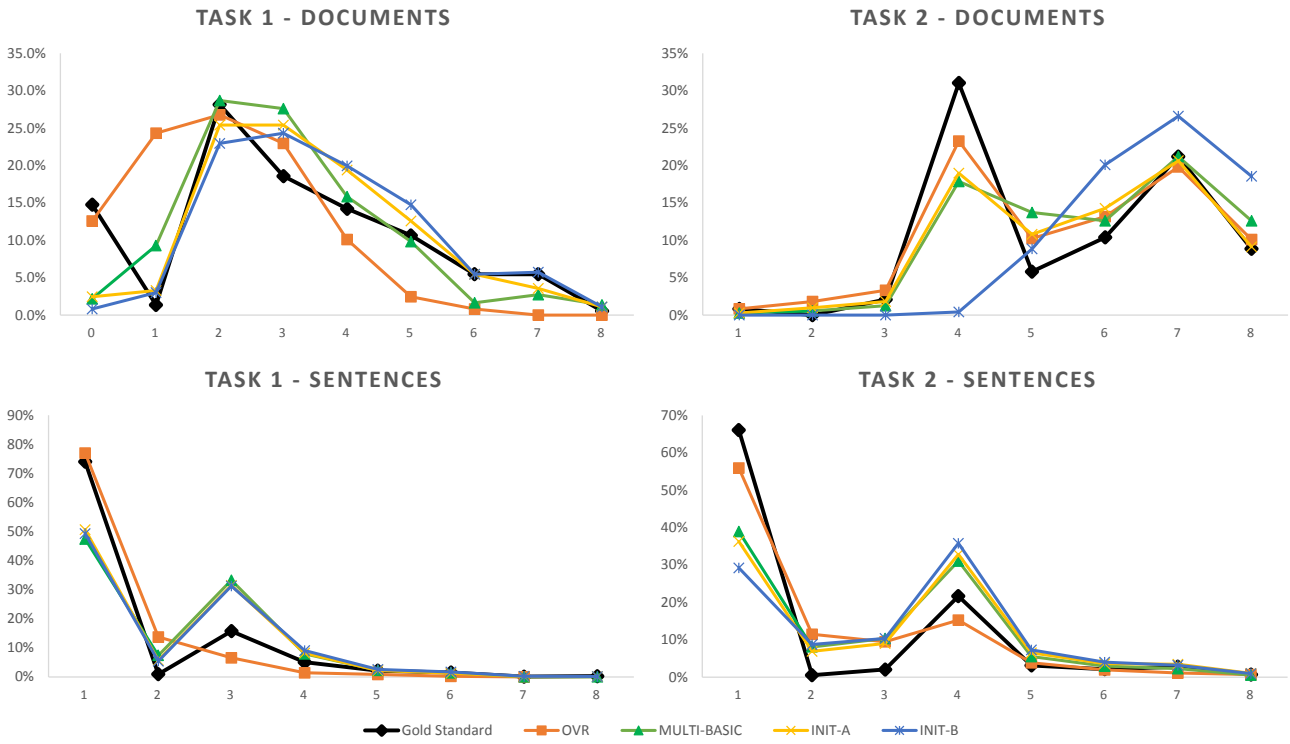

Figure 5: The distribution of instances according to the number labels per instance. The number of labels per instance (x-axis), and y-axis is the proportion of instances in the test dataset that have that number of labels. The black line indicates the distribution of the gold standard annotation (*i.e.* ground truth).

propriate subset of label co-occurrences using a certain criteria such as the frequency in the training data. For the datasets used in this paper, this was not necessary.

Overall, the results of the evaluation show that initializing the model using only label co-occurrences (INIT-A) generally produced a higher performance compared to the other models, including the random initialization of remaining hidden units in the final hidden layer (the INIT-B model) as proposed by Kurata et al. (2016). However, there was one key exception in Task 2 sentence level classification, where the one-*vs*.-rest OVR model achieved the best results.

Both variants of the initialization models investigated here achieved generally positive results when the scope of text is larger (*i.e.* documents), where there are more labels assigned per text instance. However, due to time and computational constraints, this initialization method was not fully utilized as we could only investigate its performance under a closed set of hyperparamaters for the CNN model.

It may be possible for this approach to yield even better results if further parameters are included in the CNN models (*e.g.* more filters and filter sizes). It is also important to note that collectively the one-*vs*.-rest models have much more parameters than any of the the multi-label models in our experiment setup, and therefore they have a higher capacity to capture correlations. In spite of this, the multi-label models have largely outperformed the OVR model.

8 Conclusions

There are many tasks in the biomedical domain that require the assignment of one or more labels to input text. These labels often exists within some hierarchical structure (such as a taxonomy).

The conventional approach is to use a one-*vs*.-rest classification setup: a binary classifier is trained for each label in the taxonomy or ontology where all instances not belonging to the class are considered negative examples. The main drawbacks to this approach are that dependencies between classes are not leveraged in the training and classification process, and the additional computational cost of training a classifier for each class.

We applied a new method for multi-label classification that initializes a neural network model

final hidden layer to leverage label co-occurrence. This approach elegantly lends itself to hierarchical classification.

We evaluated this approach using two hierarchical multi-label classification tasks using both sentence and document level classification. We use a baseline CNN model with a sigmoid output for each class, and a binary cross-entropy loss function. We investigated two variants of the initialization procedure. One used only co-occurrence (and hierarchical information), while the other randomly assigned random values to the remaining hidden units in the final hidden layer as proposed by Kurata et al. (2016). The experimental results for both tasks show that overall, our proposed initialization procedure (INIT-A) achieved better results than all of the the other models, with the exception of sentence-level classification in Task 2, where one-*vs*.-rest classification attained the best result. We believe that this approach shows promising potential for improving the performance for hierarchical multi-label text classification tasks.

For future work, we plan to try different initialization schemes in addition to the upper bound parameter by Glorot and Bengio (2010) that was used in the paper, and the application of this approach to other tasks and datasets such as Medical Subject headings (MeSH) text classification.

Acknowledgements

The first author is funded by the Commonwealth Scholarship Commission and the Cambridge Trust. This work is supported by Medical Research Council grant MR/M013049/1 and the Google Faculty Award. We thank Tyler Griffiths for his help in proofreading and editing this paper.

References

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2016. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics* 32(3):432–440.

Ricardo Cerri, Rodrigo C Barros, and André CPLF De Carvalho. 2014. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences* 80(1):39–56.

Guibin Chen, Deheng Ye, Erik Cambria, Jieshan Chen, and Zhenchang Xing. 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. IJCNN.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical NLP. In *Proceedings of BioNLP*.

François Chollet. 2015. Keras. https://github.com/fchollet/keras.

Wyatt T Clark and Predrag Radivojac. 2013. Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics* 29(13):i53–i61.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*. volume 9, pages 249–256.

Douglas Hanahan and Robert A Weinberg. 2000. The hallmarks of cancer. *Cell* 100(1):57–70.

Jin-Hyuk Hong and Sung-Bae Cho. 2008. A probabilistic multi-class strategy of one-vs.-rest support vector machines for cancer classification. *Neurocomputing* 71(16):3275–3281.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In *Proceedings of NAACL-HLT*. pages 521–526.

Kristin Larsson, Simon Baker, Ilona Silins, Yufan Guo, Ulla Stenius, Anna Korhonen, and Marika Berglund. 2017. Text mining for improved exposure assessment. *PloS one* 12(3):e0173132.

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.

Carolyn E Lipscomb. 2000. Medical subject headings (mesh). *Bulletin of the Medical Library Association* 88(3):265.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale multi-label text classificationrevisiting neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pages 437–452.

Sampo Pyysalo, Tomoko Ohta, and Sophia Ananiadou. 2013. Overview of the cancer genetics (CG) task of BioNLP Shared Task 2013. In *BioNLP Shared Task 2013 Workshop*.

Artem Sokolov and Asa Ben-Hur. 2010. Hierarchical classification of gene ontology terms using the gostruct method. *Journal of bioinformatics and computational biology* 8(02):357–376.

Artem Sokolov, Christopher Funk, Kiley Graim, Karin Verspoor, and Asa Ben-Hur. 2013. Combining heterogeneous data sources for accurate functional annotation of proteins. *BMC bioinformatics* 14(3):S10.

Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, pages 521–528.

Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering* 18(10):1338–1351.