All-in Text: Learning Document, Label, and Word Representations Jointly

Jinseok Nam and **Eneldo Loza Mencía** and **Johannes Fürnkranz**
Knowledge Discovery in Scientific Literature, TU Darmstadt
Knowledge Engineering Group, TU Darmstadt
Research Training Group AIPHES, TU Darmstadt

Abstract

Conventional multi-label classification algorithms treat the target labels of the classification task as mere symbols that are void of an inherent semantics. However, in many cases textual descriptions of these labels are available or can be easily constructed from public document sources such as Wikipedia. In this paper, we investigate an approach for embedding documents and labels into a joint space while sharing word representations between documents and labels. For finding such embeddings, we rely on the text of documents as well as descriptions for the labels. The use of such label descriptions not only lets us expect an increased performance on conventional multi-label text classification tasks, but can also be used to make predictions for labels that have not been seen during the training phase. The potential of our method is demonstrated on the multi-label classification task of assigning keywords from the Medical Subject Headings (MeSH) to publications in biomedical research, both in a conventional and in a zero-shot learning setting.

1 Introduction

Classification is a classical task in machine learning whose goal is to assign class labels to instances based on instances' properties. This can be seen as a learning process to identify common properties in instances and to aggregate instances, which are characterized by similar properties, in the same class. That is, classes represent commonality among instances in an abstract level. Thus, we evaluate how well the classifiers generalize to *unseen instances*. In a similar sense, evaluation can also be extended to the performance of the classifiers on *unseen labels*. For the latter, however, classification algorithms cannot work well if they exploit association patterns only between instances and labels given in the training set. This is because, in classification problems, a label is often represented by one of a fixed number of discrete values. In other words, there is no way to know how unseen labels are related to seen labels. This sort of problem is often referred to as "*zero-shot learning*" (ZSL) where a subset of labels is associated with none of training examples, but only appears among the target labels at test time (Farhadi et al. 2009; Palatucci et al. 2009). Hence, the main question in ZSL is how we can define more meaningful labels in order to improve performance of classifiers even on unseen labels.

Recently, several approaches have been proposed to address ZSL problems by making use of additional information such as attributes of labels (Lampert, Nickisch, and Harmeling 2014) and their textual information such as the labels' name (Frome et al. 2013; Socher et al. 2013; Akata et al. 2015). Such information allows for classifiers to make reasonable predictions on unseen instances associated with unseen labels, without losing generalization performance. As an example, assume that we are given a classifier trained on a collection of documents about "dogs" and "cats." What if documents about "wolves" and "lions" arrive at test time? Given the fixed label set, i.e., "dogs" and "cats," the classifier may predict the label of documents about wolves as "dogs" because it is likely that the documents about "wolves" shares more terms with ones about dogs than cats. Similarly, the documents about lions will be predicted as "cats." Let us consider a slightly different scenario that "wolves" and "lions" are also used as labels to be predicted at test time even though we did not train the classifier for such labels. Defining $A \prec B$ which means A comes before B in a ranked list, we want the classifier to yield the following ranked lists of labels for the documents about wolves: "dogs" $\prec$ "cats" $\prec$ "wolves" $\prec$ "lions," "dogs" $\prec$ "wolves" $\prec$ "cats" $\prec$ "lions," or, ideally, "wolves" $\prec$ "dogs" $\prec$ "cats" $\prec$ "lions" based on the fact that "dogs" and "wolves" belong to the same family, and under the assumption that the classifier also knows such fact learned from external resources. In other words, for the documents about wolves it is reasonable that "wolves" always precedes "lions" in label ranking based on the relationship between "dogs" and "wolves."

One way that allows classifiers to learn relationships between labels and to exploit the information for making predictions for unseen labels has been introduced in (Frome et al. 2013). This approach first represents words as d-dimensional vectors. These word embeddings are learned from large textual corpora such as Wikipedia whose vocabulary includes textual descriptions for labels such as "dogs" and "cats". In turn, representations of words corresponding to label names are used instead when labels need to be considered. As the embedding space has the interesting property that words used in similar contexts have similar representations, one is able to make reasonable predictions for unseen labels even when no prior information on them is available.

Although it sheds light on an interesting direction of ZSL, it is still problematic when we consider this method on problems where textual information of labels is quite complex to be converted into words by looking up in the dictionary. To circumvent this problem, one can make the assumption that each label has its own description in textual format. Then, such descriptions can be represented by *tf-idf* as in (Elhoseiny, Saleh, and Elgammal 2013). For example, "dog" in Wikipedia is described as follows:

> The domestic dog (*Canis lupus familiaris* or *Canis familiaris*) is a domesticated canid which has been selectively bred for millennia for various behaviors, sensory capabilities, and physical attributes. . . .

Furthermore, it is worth noting that learning word representations is independent of the training data in (Frome et al. 2013). If instances are also in textual format, we may further exploit word embeddings by finding a joint space of all available information such as word sequence patterns in both instances and label descriptions, and association patterns between instances and labels.

Hence, in this paper, we aim at learning document, label, and word representations from such textual information where labels descriptions and documents share the same word vocabulary, as well as association patterns between documents and labels. This joint learning scheme allows us to infer representations for unseen labels and to obtain better classification systems in terms of generalization performance on both unseen instances and labels.

2 Problem Statement

In the following we will define a set of notations which will be used throughout this work. Assume that we are given a vocabulary of V words $\mathcal{W} = \{1, 2, \cdots, V\}$, a set of L labels $\mathcal{C}_s = \{1, 2, \cdots, L\}$, and a set of N training examples $\mathcal{D} = \{(\mathcal{T}_x^{(n)}, \mathcal{Y}^{(n)})_{n=1}^N\}$ where $\mathcal{T}_x^{(n)} = \{w_1^{(x)}, w_2^{(x)}, \cdots, w_{M_n}^{(x)}\}$ denotes a sequence of M_n words $w \in \mathcal{W}$, and $\mathcal{Y}^{(n)} = \{y_1, y_2, \cdots, y_{Q_n}\}$ a set of Q_n relevant labels $y \in \mathcal{C}_s$ for the n-th training example. Each label $y_l \in \mathcal{C}_s$ has its own description $\mathcal{T}_y^{(l)} = \{w_1^{(y)}, w_2^{(y)}, \cdots, w_{M_l}^{(y)}\}$ consisting of M_l words. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \in \mathbb{R}^{k \times N}$, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_L\} \in \mathbb{R}^{k \times L}$ and $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_V\} \in \mathbb{R}^{k \times V}$ be document, label, and word representations, respectively. For example, $\mathbf{x}_1$ corresponds to the k-dimensional vector for the document indexed 1 in $\mathcal{D}$, i.e., $\mathcal{T}_x^{(1)}$.

In this work, we examine our hypothesis on a multi-label text classification dataset where $|\mathcal{Y}^{(n)}| \geq 1$ for all n. Given multiple labels per document, our task is to learn a ranking function which yields higher similarity scores between a document and its relevant labels than ones between a document and irrelevant labels. More formally, the objective is to learn a ranking function $f : (\mathbf{x}, \mathbf{y}) \to \mathbb{R}$ such that $f\left(\mathbf{x}, \mathbf{y}_{y_p}\right) > f\left(\mathbf{x}, \mathbf{y}_{y_n}\right)$ where $y_p \in \mathcal{Y}$ and $y_n \in \overline{\mathcal{Y}}$.

At test time we have a set of *unseen* labels $\mathcal{C}_u = \{L + 1, L + 2, \cdots, L + L_u\}$ and each unseen label $y_l^* \in \mathcal{C}_u$ also has its description $\mathcal{T}_{y^*}^{(l)}$.

3 Method

In this section, we describe how to learn representations of both documents and labels jointly from their textual description in a way that a document and its relevant labels yield higher similarity scores in the joint embedding space.

3.1 Documents and Labels as Word Sequences

As for documents, i.e., instances represented by sequences of words, we can also deal with labels as instances of word sequences, provided they have textual descriptions. Based on the assumption that a representation of such an instance should contain *global* information on its description, one can learn fixed-size vector representations for documents and labels while learning a *local* predictor of a word given its context in the textual description (Le and Mikolov 2014).

Given the training document set $\mathcal{K}_X = \{\mathcal{T}_x^{(n)} | 1 \leq n \leq N\}$, firstly, we show how to learn representations for a *document* and individual *words*, respectively. For convenience, we will drop n from both $\mathcal{T}_x^{(n)}$ and $\mathbf{x}_n$ when it is not confusing. Note that the document representation $\mathbf{x}$ is a set of learnable parameters as well as the word representations. The objective function is to maximize the probability of predicting a word at position t in $\mathcal{T}_x$ given its $c - 1$ surrounding words and the document representation $\mathbf{x}$:

$$p(w_t|\mathbf{w}_{-t}, \mathbf{x}) = \frac{\exp({\mathbf{u}'}_{w_t}^T \hat{\mathbf{u}}_{w_t})}{\sum_{v=1}^V \exp({\mathbf{u}'}_v^T \hat{\mathbf{u}}_{w_t})} \tag{1}$$

where $\mathbf{u}'_{w_t}$ is the ck-dimensional vector for an output word w_t, and $\hat{\mathbf{u}}_{w_t}$ denotes the *context* representation of the output word, which is a concatenation of representations for the context words $\mathbf{w}_{-t} = \{w_{t-(c-1)/2}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+(c-1)/2}\}$ and the document representation $\mathbf{x}$ defined as

$$\hat{\mathbf{u}}_{w_t} = \left[\mathbf{x}, \mathbf{u}_{w_{t-(c-1)/2}}, \cdots, \mathbf{u}_{w_{t+(c-1)/2}}\right] \in \mathbb{R}^{ck}. \tag{2}$$

Here, $\hat{\mathbf{u}}_{w_t}$ can be interpreted as a combination of *global* (i.e., $\mathbf{x}$) and *local* (i.e., $\left[\mathbf{u}_{w_{t-(c-1)/2}}, \cdots, \mathbf{u}_{w_{t+(c-1)/2}}\right]$) context information of a word w_t in $\mathcal{T}_x$. Instead of using the softmax in Eq. 1 directly, we use its approximation, namely *negative sampling* (Mikolov et al. 2013):

$$\begin{aligned} &\log p\left(w_t|\mathbf{w}_{-t}, \mathbf{x}\right) \\ &\approx \log \sigma({\mathbf{u}'}_{w_t}^T \hat{\mathbf{u}}_{w_t}) + \sum_{i=1}^{\kappa} \mathbb{E}_{P_n(w)}\left[\log \sigma({\mathbf{u}'}_{w_i}^T \hat{\mathbf{u}}_{w_t})\right] \end{aligned} \tag{3}$$

where $\sigma(x)$ is the sigmoid function, κ is the number of negative samples, and $P_n(w)$ is the unigram distribution raised to the power of $3/4$.

Then, we optimize both $\mathbf{X}$ and $\mathbf{U}$ in a way of maximizing the average log probability over all words in documents $\mathcal{K}_X$ as follows

$$\begin{aligned} &\mathcal{L}_X\left(\Theta_X; \mathcal{K}_X\right) \\ &= \sum_{n=1}^N \frac{1}{|\mathcal{T}_x^{(n)}|} \sum_{t=1}^{|\mathcal{T}_x^{(n)}|} -\log p(w_{t,n}|\mathbf{w}_{-t,n}, \mathbf{x}_n) \end{aligned} \tag{4}$$

where $\Theta_X = \{\mathbf{X}, \mathbf{U}, \mathbf{U}'\}$. Similarly, one can learn $\mathbf{Y}$ for the label descriptions $\mathcal{K}_Y = \{\mathcal{T}_y^{(l)} | 1 \leq l \leq L\}$ and $\mathbf{U}$:

$$\mathcal{L}_Y(\Theta_Y; \mathcal{K}_Y) = \sum_{l=1}^{L} \frac{1}{|\mathcal{T}_y^{(l)}|} \sum_{t=1}^{|\mathcal{T}_y^{(l)}|} -\log p(w_{t,l} | \mathbf{w}_{-t,l}, \mathbf{y}_l) \tag{5}$$

where $\Theta_Y = \{\mathbf{Y}, \mathbf{U}, \mathbf{U}'\}$.

3.2 Joint Embeddings

So far we have discussed how to learn document, label and word representations jointly from textual description of documents and labels. Once we learn the document representations $\mathbf{X}$ and the label representations $\mathbf{Y}$, they are assumed to be *global* representations for their textual description. In that case, modeling the relationship between documents and labels is disregarded. However, since our goal in multi-label classification tasks is to make relevant labels distinguishable from irrelevant labels for a given instance, we learn a ranking function to place relevant labels at the top of a ranking of labels by similarity scores w.r.t. a given instance.

Defining the $k \times k$ matrix $\mathbf{W}$, the bilinear function $f(\mathbf{x}, \mathbf{y})$ is written as

$$f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{W} \mathbf{y}. \tag{6}$$

By using the bilinear function $f(\mathbf{x}, \mathbf{y})$, we can compute the rank of $y_p \in \mathcal{Y}$ with respect to $\mathbf{x}$ as sum of the number of incorrectly ranked pairs as follows

$$\Psi(\mathbf{x}, y_p) = \sum_{y_n \in \bar{\mathcal{Y}}} \mathbb{I}\left[f\left(\mathbf{x}, \mathbf{y}_{y_p}\right) \leq f\left(\mathbf{x}, \mathbf{y}_{y_n}\right)\right] \tag{7}$$

where $\mathbb{I}[\cdot]$ takes 1 if its argument is true otherwise 0. The overall loss is, then, the sum of the average rank of relevant labels for a document representation over the training set:

$$\mathcal{L}_r(\Theta_J; \mathcal{D}) = \sum_{n=1}^{N} \frac{1}{|\mathcal{Y}^{(n)}|} \sum_{y_p \in \mathcal{Y}^{(n)}} \Psi(\mathbf{x}, u_p) \tag{8}$$

where $\Theta_J = \{\mathbf{X}, \mathbf{Y}, \mathbf{W}\}$.

As it is difficult to optimize the loss function in Eq. 8 directly, one can consider instead the *Weighted Approximate Rank Pairwise* (WARP) loss (Weston, Bengio, and Usunier 2011), which uses an *approximation* of Eq. 7 given by

$$\Psi^*(\mathbf{x}, y_p) = \sum_{y_v \in \mathcal{V}_{y_p}} w(y_p)\left[m - f(\mathbf{x}, \mathbf{y}_{y_p}) + f(\mathbf{x}, \mathbf{y}_{y_v})\right]_+ \tag{9}$$

where $w(y_p)$ is a weight of the positive label y_p, $[x]_+$ outputs x if $x > 0$ otherwise 0, $m \in \mathbb{R}$ denotes a margin, and $\mathcal{V}_{y_p}$ is the set of labels defined by

$$\mathcal{V}_{y_p} = \{y_n | (m + f(\mathbf{x}, \mathbf{y}_{y_n})) \geq f(\mathbf{x}, \mathbf{y}_{y_p}), \forall y_n \in \bar{\mathcal{Y}}\}. \tag{10}$$

For a weight $w(y_p)$, a truncated harmonic function can be used as follows

$$w(y_p) = \sum_{i=1}^{r(y_p)} \frac{1}{i} \tag{11}$$

where $r(y_p) = \sum_{y_v \in \mathcal{V}_{y_p}} \mathbb{I}\left[m + f(\mathbf{x}, \mathbf{y}_{y_v}) \geq f(\mathbf{x}, \mathbf{y}_{y_p})\right]$ is the rank of y_p. Due to computational cost of Eq. 11, which allows us to optimize precision at the rank of y_p (Usunier, Buffoni, and Gallinari 2009), it is further approximated by

$$w(y_p) \approx \left\lfloor \frac{L - |\mathcal{Y}|}{S} \right\rfloor \tag{12}$$

where S is the number of samples drawn uniformly from $\bar{\mathcal{Y}}$ until a label $y_v \in \mathcal{V}_{y_p}$ is sampled. By substituting the ranking loss in Eq. 8 by Eq. 9, we obtain the WARP loss:

$$\mathcal{L}_w(\Theta_J; \mathcal{D}) = \sum_{n=1}^{N} \frac{1}{|\mathcal{Y}^{(n)}|} \sum_{y_p \in \mathcal{Y}^{(n)}} \Psi^*(\mathbf{x}, y_p). \tag{13}$$

Algorithm 1: Training *AiTextML*

```
input : $\mathcal{D} = \{(\mathcal{T}_x^{(n)}, \mathcal{Y}^{(n)})_{n=1}^{N}\}$,
        $\mathcal{K}_Y = \{\mathcal{T}_y^{(l)} | 1 \leq l \leq L\}$
output: $\Theta = \{\mathbf{U}, \mathbf{U}', \mathbf{X}, \mathbf{Y}, \mathbf{W}\}$
do
    for $n = 1$ to $N$ do
        $\mathcal{V}^* \leftarrow \emptyset$    // violation labels set
        foreach $y_p \in \mathcal{Y}^{(n)}$ do
            $S \leftarrow 0$
            $pos \leftarrow f(\mathbf{x}_n, \mathbf{y}_{y_p})$
            do
                $S \leftarrow S + 1$
                pick $y_n$ from $\{1, \cdots, L\}$ at random
                $neg \leftarrow f(\mathbf{x}_n, \mathbf{y}_{y_n})$
                if $m + neg \geq pos$ then
                    $\mathcal{V}^* \leftarrow \mathcal{V}^* \cup y_n$
                    update $\Theta_J$ using Eq. 13
                    break
            while $m + neg \leq pos$ and $S < L - |\mathcal{Y}|$
        foreach $w_t \in |\mathcal{T}_x^{(n)}|$ do
            update $\Theta_X$ using Eq. 4
        foreach $l \in \{\mathcal{Y}^{(n)} \cup \mathcal{V}^*\}$ do
            foreach $w_t \in |\mathcal{T}_y^{(l)}|$ do
                update $\Theta_Y$ using Eq. 5
while until termination conditions are met
```

3.3 Putting It All Together

Our goal is to learn representations for documents, labels, and words, which are all in textual format, jointly to improve the generalization performance of our proposed method to unseen labels as well as to seen ones on multi-label text classification datasets. We call this method All-in Text Multi-label Learner (*AiTextML*). The goal is achieved by combining the losses regarding document and label representations from word sequences in Eqs. 4 and 5, and the WARP loss, i.e., Eq. 13. Thus, the objective is

$$\begin{gathered} \mathcal{L}(\Theta; \mathcal{D}, \mathcal{K}_Y) = \alpha \mathcal{L}_w + \beta \mathcal{L}_X + \gamma \mathcal{L}_Y \\ \text{s.t. } \alpha + \beta + \gamma = 1 \end{gathered} \tag{14}$$

Table 1: Statistics of the BioASQ dataset

# training examples (N)	6,692,815
# validation examples (N_v)	100,000
# test examples (N_t)	4,912,719
# words (V)	528,156
# *seen* labels (L)	23,669
# *unseen* labels (L_u)	2,435
Avg. # of relevant seen labels per training example	10.83
# test examples that have unseen labels	432,703
Avg. ratio of relevant unseen labels in the test set	10.31%

where $\Theta = \{\mathbf{U}, \mathbf{U}', \mathbf{X}, \mathbf{Y}, \mathbf{W}\}$ denotes the set of parameters which are randomly initialized, and the control parameters α, β, γ determine the impact of the WARP loss $\mathcal{L}_w$ and the representation learning losses $\mathcal{L}_X$ and $\mathcal{L}_Y$ to the total loss $\mathcal{L}$. We use stochastic gradient descent (SGD) with a fixed learning rate η for all time steps τ to update the parameters Θ given a training example indexed n at a time:

$$\Theta_{\tau+1} := \Theta_\tau - \eta \frac{\partial \mathcal{L}(\Theta_\tau; \mathcal{T}_X^{(n)}, \mathcal{Y}^{(n)}, \mathcal{K}_Y)}{\partial \Theta_\tau}. \tag{15}$$

The pseudo-code of our proposed method is shown in Alg. 1.

3.4 Inference on Unseen Documents and Labels

As shown in the previous sections, our proposed method needs document and label representations to be estimated as parameters from word sequences. The same holds for unseen data points at test time. Consider that we are given a test set $\mathcal{D}^* = \{(\mathcal{T}_{x^*}^{(n)}, \mathcal{Y}^{*(n)})\}_{n=1}^{N_t}$, and that some of labels do not appear in the training set such that $y_{(\cdot)}^* \in \{L+1, L+2, \cdots, L+L_u\}$ where L_u is the number of unseen labels. To make predictions on unseen documents w.r.t. unseen labels as well, we initialize $\mathbf{X}^* = \{\mathbf{x}_1^*, \mathbf{x}_2^*, \cdots, \mathbf{x}_{N_t}^*\}$ and $\mathbf{Y}^* = \{\mathbf{y}_1^*, \mathbf{y}_2^*, \cdots, \mathbf{y}_{L_u}^*\}$ randomly for unseen documents and labels, respectively. In turn, we define only $\mathbf{X}^*$ and $\mathbf{Y}^*$ as trainable parameters for the *AiTextML* model on the test set $\mathcal{D}^*$ while all the other parameters $\{\mathbf{U}, \mathbf{U}', \mathbf{X}, \mathbf{Y}, \mathbf{W}\}$ are kept fixed. At inference time, we use the same control parameters α, β and number of parameter updates used in the training phase. To prevent learning $\mathbf{X}^*$ and $\mathbf{Y}^*$ from document-label association patterns in $\mathcal{D}^*$, we set γ to 0.

Note that as unseen document representations $\mathbf{x}^*$ and unseen label representations $\mathbf{y}^*$ are independent of each other, we can easily parallelize this inference stage.

4 Experimental Setup

4.1 Dataset

We use the BioASQ Task 3a dataset, a collection of scientific publications in biomedical research, to examine our proposed method.[1] It contains about 12 million publications, each of which is associated with around 11 descriptors on average out of 27,455, which come from the Medical Subject Headings (MeSH) hierarchy.[2] We removed 1003 descriptors from the MeSH hierarchy because they do not have textual descriptions as well as 348 descriptors not appearing in the BioASQ Task 3a dataset. We split the dataset by year so that the training set includes all papers by 2004 and the rest of papers published between 2005 and 2015 belongs to the test set. Thus, descriptors introduced to the MeSH hierarchy after 2004 can be considered as unseen labels. 100,000 papers before 2005 were randomly sampled and set aside as the validation set for tuning hyperparameters. Since we split the dataset by year, 2,435 labels in the test set do not appear in the training set. About 10% of test examples contain such unseen labels in their target label set. The ratio of unseen labels in the target label set of the test data is 10.31%.

We applied minimal preprocessing to documents and label descriptions; tokenization and replacement of numbers and rare words to special tokens, e.g., *NUM* and *UNK*. The word vocabulary was built according to the word frequency in the training documents, for which words occurring more than 10 times were chosen. The statistics on the dataset used are summarized in Table 1.

4.2 Baseline

Since no work has been reported yet in this line of research to our best knowledge, we compare *AiTextML* with the same model using fixed $\gamma = 0$ in Eq. 14. That is, our baseline also optimizes the WARP loss. However, our baseline considers learning representation of documents and words simultaneously, whereas Wsabie in (Weston, Bengio, and Usunier 2011) uses fixed feature representations for instances. Hence, our baseline is also able to learn feature representations and can be seen as an extension of Wsabie. Unlike conventional multi-label learning algorithms, Wsabie scales well on large-scale datasets in terms of both the number of training examples and labels, and performs comparably even in standard benchmark datasets for multi-label text classification (Nam et al. 2015).

4.3 Evaluation Measures

We report the performance of our proposed method using three measures: rank loss, average precision and one-error (Schapire and Singer 2000). The rank loss measures the quality of label ranking given by

$$\mathrm{RL}(\mathbf{x}, \mathcal{Y}) = \frac{1}{|\mathcal{Y}||\overline{\mathcal{Y}}|} \sum_{\substack{(y_p, y_n) \\ \in \mathcal{Y} \times \overline{\mathcal{Y}}}} \mathbb{I}\left[f(\mathbf{x}, \mathbf{y}_{y_p}) \leq f(\mathbf{x}, \mathbf{y}_{y_n})\right] \tag{16}$$

which has the same form except for the normalization factor $|\overline{\mathcal{Y}}|$ with the ranking function in Eq. 7. We can compute average precision at the position of relevant labels:

$$\mathrm{AvgPr}(\mathbf{x}, \mathcal{Y}) = \frac{1}{|\mathcal{Y}|} \sum_{\substack{(y_p, y_t) \\ \in \mathcal{Y} \times \mathcal{Y}}} \frac{\mathbb{I}\left[f(\mathbf{x}, \mathbf{y}_{y_p}) \geq f(\mathbf{x}, \mathbf{y}_{y_t})\right]}{\Psi(\mathbf{x}, y_p) + 1}. \tag{17}$$

[1] http://www.bioasq.org/participate/data

[2] https://www.nlm.nih.gov/mesh/introduction.html

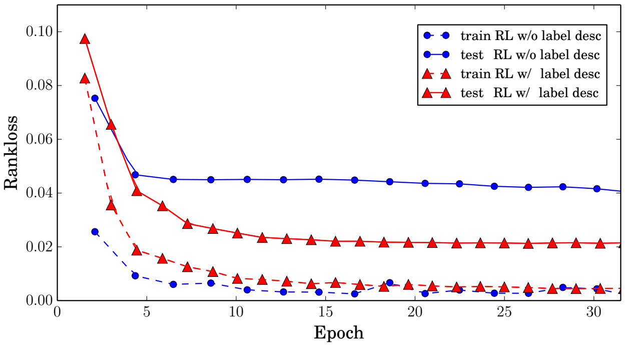

Figure 1: Effect of learning from label descriptions in terms of rank loss on the BioASQ dataset w.r.t. the *seen* labels. The rank loss was estimated on randomly sampled 10,000 training examples and on a fixed subset of 10,000 test examples every 60 mins in the course of training, indicated by markers.

The one-error loss accounts for the accuracy of a label ranked at the top defined as

$$\mathrm{OneErr}(\mathbf{x}, \mathcal{Y}) = 1 - \mathbb{I}\left[\underset{i\in\{1\cdots L\}}{\arg\max} f(\mathbf{x}, \mathbf{y}_i) \in \mathcal{Y}\right]. \quad (18)$$

In addition to the commonly used measures in multi-label classification, we evaluate the performance of models per label given ranked lists of labels. Let us define a set of document indices which are associated with label size of s as $\mathcal{A}_s$, and $\varphi(y)$ as the size of a label y. For example, if a label y appears only in a single training document, $\varphi(y) = 1$. Label-based average rank (AvgRank) with respect to label size s is given by:

$$\mathrm{AvgRank}(s) = \frac{1}{\mathcal{Z}_n^s} \sum_{n\in\mathcal{A}_s} \sum_{y_p\in\mathcal{Y}^{(n)}} \left[\Psi(\mathbf{x}_n, y_p) + 1\right]_{\varphi(y_p)=s} \quad (19)$$

where $\mathcal{Z}_n^s = |\mathcal{A}_s| \sum_{y_p\in\mathcal{Y}^{(n)}} \mathbb{I}\left[\varphi(y_p) = s\right]$ and $[x]_{\varphi(y_p)=s}$ outputs x if $\varphi(y_p) = s$ is true otherwise 0.

5 Experiments

We used the validation set to set our hyperparameters as follows: the number of negative samples $\kappa = 5$, the dimensionality of all representations 100, the size of the context window $c = 5$, learning rate $\eta = 0.025$, margin $m = 0.1$, and the control variables $\alpha = 1/3, \beta = 1/3, \gamma = 1/3$. For the baseline, different control parameters $\alpha = 1/3, \beta = 2/3, \gamma = 0$ were used, but the rest of the hyperparameters were same with our proposed method. Unless we specify otherwise, the hyperparameter settings are used throughout all experiments. In order to prevent overfitting, we impose constraints on norm of document, label and word vectors such that $\|\mathbf{u}_i\|_2 \leq 1, i \in \{1, \cdots, V\}$, $\|\mathbf{x}_d\|_2 \leq 1, d \in \{1, \cdots, N\}$, and $\|\mathbf{y}_l\|_2 \leq 1, l \in \{1, \cdots, L\}$. We performed all experiments on a machine with two Intel Xeon E5-2670 CPUs and 32GB of memory.

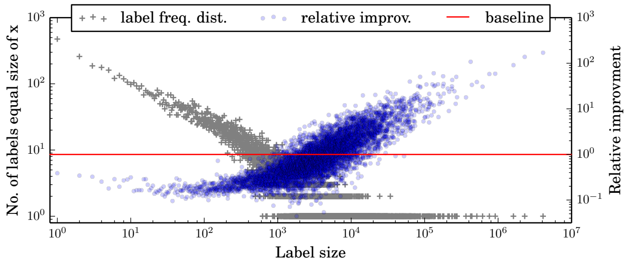

Figure 2: Label frequency distribution and relative improvement over the baseline with respect to label size.

Table 2: Comparison of *AiTextML* to the baseline w.r.t. *seen* labels. The *AiTextML* model was trained for the same amount of time (24 hrs) as the baseline. The numbers in the parentheses following the methods correspond to the control parameters (α, β, γ) in Eq. 14.

	RL	AvgPr	OneErr
Baseline $\left(\frac{1}{3}, \frac{2}{3}, 0\right)$	0.05217	**0.36645**	0.41728
AiTextML $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$	**0.03544**	0.32786	**0.25992**

5.1 Effect of Label Descriptions

We carried out experiments to compare the models which learns purely from the association patterns and the other which learn from label descriptions as well as the association patterns. As can be seen in Fig. 1, learning from label descriptions improves the generalization performance of our method. Indeed, rank loss on the training set of the model without learning from label descriptions is even lower than that of the model trained on label descriptions. In contrast to the baseline, *AiTextML* achieves better rank loss scores on the test set. This shows that label descriptions help *AiTextML* prevent from overfitting. Since *AiTextML* learns label representations not only from the association patterns, but also textual description of labels, it takes more time for a single iteration indeed under the same hyperparameter settings.

Once having trained *AiTextML* and the baseline for 50 epochs, we evaluated two models on the full set of test examples. We observed that *AiTextML* outperforms substantially the baseline in terms of rank loss and one-error, which tells us learning from label descriptions plays an important role for the improvements. However, AvgPr of our proposed method rather decreases compared to the baseline. Note that our objective measure in the optimization corresponds to ranking. The results are shown in Table 2. It is often the case that label frequency distribution in real world multi-label text datasets follows a *power law* as shown in Fig. 2, which means, informally, there are few frequent labels, but many infrequent ones. This property makes it difficult for classifiers to generalize well to unseen instances if they have rare labels in their target labels since a classifiers tend to overfit rare labels.

In order to take a closer look at the source of improvements, we comapared both the baseline and our proposed

Table 3: Nearest neighbors for given *unseen* labels in seen and unseen label representations.

	Tundra	Night Vision	Hope
Seen labels	Genetic Speciation	Halorhodopsins	Adult Children
	Arcidae	Fluorophotometry	World War II
	Secernentea	Arthropod Compound Eye	Healthy Volunteers
	Biological Extinction	Retinoscopes	World War I
	Wetlands	Color Vision	Health Status Disparities
Unseen labels	Grassland	Retinal Photoreceptor Cell Outer Segment	Time-to-Treatment
	Permafrost	Mesopic Vision	Anatomists
	Click Chemistry	Plant Photoreceptors	Pragmatic Clinical Trials as Topic
	Ponds	Rod-Cone Interaction	Secondary Care
	Cambium	Bleaching Agents	Historically Controlled Study

method in terms of AvgRank. Fig. 2 shows that *AiTextML* performs significantly better than the baseline for frequent labels, whereas its performance on rare labels is worse than the baseline. Our model learns more often from descriptions of frequent labels in a way that their representations are effective in predicting a next word given its context and maximizing similarity scores to the documents that they belong to as well. Due to the fact that *AiTextML* focuses more on frequent labels, average ranks rare labels are rather ignored which results in lower average precision.

5.2 Unseen Label Representations

We demonstrate the quality of unseen label representations by listing nearest neighbors in both *seen* and *unseen* label spaces to selected unseen labels, shown in Table 3. For example, given a query "Tundra," we have "Genetic Speciation," "Biological Extinction," and "Wetlands" as similar labels from the seen label set, which are somehow related to environmental danger in the tundra. "Grassland" from the unseen label set is another type of biomes which is often used to contrast different characteristics of "Tundra." "Permafrost" and "Ponds" are also related labels to "Tundra" when a paper discusses climate changes and their effects in the tundra. Such relationships can be also found for the unseen label "Night Vision."

In contrast, there is no clear relationship between the unseen query label "Hope" and both seen and unseen labels. This is because such a label has a very short description and unclear terms are used in the description. For example, "Hope" is described as "Belief in a positive outcome."

5.3 Zero-Shot Prediction

One of the promising aspects of our proposed method is the capability of learning unseen label representations from their descriptions. About 400,000 test examples have 1$\sim$2 unseen labels in their target label sets on average as shown in Table 1. Without using the inference step and the joint space embedding, a reasonably straightforward solution to obtain unseen label representations is averaging embeddings of words which occur in textual description of labels including their name. For label names, we applied the same preprocessing pipeline used for the documents. For example, if we have an unseen label "1918-1919 Influenza Pandemic," it is replaced with "NUM-NUM influenza pandemic" and then its representation is determined by the averaged representations of three words "NUM-NUM," "influenza," and "pandemic." We use a special token "UNK" when a word cannot be found in the vocabulary. Also, the norm of unseen label representations is scaled to 1. Instead of learning such word embeddings independently of our task, we used word embeddings of the baseline and *AiTextML* in Sec. 5.1. Note that our baseline has the same architecture and number of parameters for *AiTextML*, but does not learn from label descriptions.

Table 4: Comparison of *AiTextML*, which represents unseen labels by the inference step, to averaging of embeddings for words in label names or descriptions on the zero-shot task. For averaging words in the textual information, we use the word embeddings from the baseline and the *AiTextML* model.

	RL	AvgPr	OneErr
Baseline avg (names)	0.50225	0.00317	0.99969
Baseline avg (desc.)	0.48812	0.00375	0.99946
AiTextML avg (names)	0.52335	0.00290	0.99979
AiTextML avg (desc.)	0.52890	0.00388	0.99941
AiTextML inf (desc.)	**0.21622**	**0.02665**	**0.98608**

We compare the proposed method with four possible combinations of two word embeddings from the baseline and *AiTextML*, and two textual information sources to be used for representing unseen labels, i.e., names and descriptions. As can be seen in Table 4, *AiTextML*, which *infers* unseen label representations from textual descriptions, outperforms the baseline models for estimating unseen label representations by averaging over representations for words appearing in either label names or descriptions. Moreover, using the averaged word embeddings from label descriptions does not achieve relevant improvements over using only the label names. In other words, when we consider the word embeddings to obtain unseen label representations, using label descriptions seems to be a better choice than label names. However, the gain is not comparable to what our proposed method achieves. This shows that the inference step for unseen label representations in our proposed method plays an important role for yielding more useful information than given by the average of word embeddings in this task.

6 Discussion

We have presented a framework for learning document, label, and word representations jointly to leverage shared information available in textual format. This allows not only to make better predictions w.r.t. seen labels, but also produces better representations for unseen labels in a zero-shot learning setting. In particular, we could show that our methods outperforms a baseline approach which simply averages representations of all words in either the label names or the label descriptions.

Our objective in this work is to jointly learn document, label and word representations to exploit shared information, and we demonstrated *AiTextML* only on textual data. However, we note that the label representation learning part can be also applied to other domains such as object classification in images under the ZSL setting instead of defining attributes for unknown labels. A major limitation when considering our proposed method in learning label representations is the availability of label descriptions. If a dataset does not have such label descriptions, one can make use of external knowledge resources such as Wikipedia to construct the label description set. For example, the first sentence or paragraph in Wikipedia articles contain very general terms for describing facts of interest.

Finally, we would like to highlight the key differences between our proposed method and the approaches where label names are used to obtain unseen label representations. The principle of *AiTextML* is more general because we can easily and efficiently add representations for unseen labels to the model by the inference step under the assumption that label descriptions consist of general terms. If words in label names are out of the vocabulary, we need to handle them more carefully because label names are rather short in general and such information loss occur frequently, which often leads to inaccurate unseen label representations in the ZSL task. Furthermore, whereas label representations by using their names provide only a good starting point for label embeddings, the proposed method allows us to obtain improved label rankings on test instances as well by learning all representations jointly in conjunction with label descriptions in the whole training process.

Acknowledgments

The authors would like to thank the anonymous reviewers for their feedback. This work has been supported by the German Institute for Educational Research (DIPF) under the Knowledge Discovery in Scientific Literature (KDSL) program, and the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1."

References

Akata, Z.; Reed, S.; Walter, D.; Lee, H.; and Schiele, B. 2015. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2927–2936.

Elhoseiny, M.; Saleh, B.; and Elgammal, A. 2013. Write a classifier: Zero shot learning using purely textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, 2584–2591.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1778–1785.

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; and Mikolov, T. 2013. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*. 2121–2129.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):453–465.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*, 1188–1196.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

Nam, J.; Loza Mencía, E.; Kim, H. J.; and Fürnkranz, J. 2015. Predicting unseen labels using label hierarchies in large-scale multi-label learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, 102–118.

Palatucci, M.; Pomerleau, D.; Hinton, G. E.; and Mitchell, T. M. 2009. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems*. 1410–1418.

Schapire, R., and Singer, Y. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning* 39(2-3):135–168.

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, 935–943.

Usunier, N.; Buffoni, D.; and Gallinari, P. 2009. Ranking with ordered weighted pairwise classification. In *Proceedings of the International Conference on Machine Learning*, 1057–1064.

Weston, J.; Bengio, S.; and Usunier, N. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2764–2770.