Du et al. BMC Bioinformatics 2018, 19(Suppl 20):502
https://doi.org/10.1186/s12859-018-2534-2

BMC Bioinformatics

RESEARCH Open Access

Biomedical semantic indexing by deep neural network with multi-task learning

Yongping Du, Yunpeng Pan*, Chencheng Wang and Junzhong Ji

From IEEE International Conference on Bioinformatics and Biomedicine 2017
Kansas City, MO, USA. 13-16 November 2017

Abstract

Background: Biomedical semantic indexing is important for information retrieval and many other research fields in bioinformatics. It annotates biomedical citations with Medical Subject Headings. In face of unbalanced category distribution in the training data, sampling methods are difficult to apply for semantic indexing task.

Results: In this paper, we present a novel deep serial multi-task learning model. The primary task treats the biomedical semantic indexing as a multi-label text classification issue that considers the relations of the labels. The auxiliary task is a regression task that predicts the MeSH number of the citation and provides hints for the network to make it converge faster. The experimental results on the BioASQ-Task5A open dataset show that our model outperforms the state-of-the-art solution "MTI", proposed by the US National Library of Medicine. Further, it not only achieves the highest precision among all the solutions in BioASQ-Task5A but also has faster convergence speed compared with some naive deep learning methods.

Conclusions: Rather than parallel in an ordinary multi-task structure, the tasks in our model are serial and tightly coupled. It can achieve satisfied performance without any handcrafted feature.

Keywords: Multi-label classification, Biomedical semantic indexing, Data mining, Natural language processing, Multi-task learning, Word embedding

Background

In order to index citations in MEDLINE, a life science and biomedicine journal database, National Library of Medicine (NLM) developed Medical Subject Headings (MeSH). Citations indexed by MeSH have been applied in fields such as query expansion [1], MEDLINE document clustering [2], enhancing search strategies for physical therapy [3] and so on. There are 28,472 MeSH main headings by 2017 [4]. Currently, the indexing work is performed by a group of qualified NLM staff, given the full text of each citation in MEDLINE. The task is becoming more and more tough on account of the annually increasing number of citations in MEDLINE (869,666 in 2016, and has approximately 8% increase over 2015 [5]). This fact leads to the manual semantic indexing task to be very inefficient and financially expensive. For example, the average cost of indexing each citation was reported to be around $9.4 [6].

The main challenge of the BioASQ task can be concluded as follows.

Insufficient information

The participants are only provided with the name of the journal where the citations were published, the titles and the abstracts of the citations due to the limit of authority. By contrast, MeSH indexing experts of NLM have the full articles. Apparently, there are lots of useful information in the full article, which is not available to the participants.

The large amount of MeSH

There are 28,472 MeSH by 2017. On the contrary, the average number of MeSH in each citation is 13 [6], thus there are much more negative labels than

* Correspondence: pypmemorypool@gmail.com
Faculty of Information Technology, Beijing University of Technology, Beijing, China

positive labels for each citation which increase the difficulty of indexing.

Unbalanced distribution of MeSH in the MEDLINE

According to the research of Ke Liu et al. [6], the most frequent MeSH "Human" appears in 8,152,852 citations, while the 25,000th frequent MeSH "Pandanaceae" appears only in 31 citations in total 12,504,999 MEDLINE citations. As a result, there are not enough positive samples to learn the correct assignment of the infrequent MeSH.

Many research works have addressed the problem of biomedical semantic indexing by a wide variety of methods, and the most recent powerful methods have mainly used machine learning methods.

For example, the "Medical Text Indexers" (MTI) [7] of NLM annotated biomedical citations with Unified Medical Language System (UMLS) [8] using MetaMap [9]. It used Restrict-to-MeSH approach and the k-Nearest Neighbor (k-NN) algorithm. MTI is one of the most advanced method for indexing biomedical citations. It is also the baseline solution of BioASQ challenge task A [10], an international competition for automatically annotating new MEDLINE citations with MeSH. Liu et al. [6] proposed model "MeSHLabeler" which extracted several different features: the result of a MeSH classifier, the scores from the nearest neighbor citations, the MeSH and their synonyms directly found in the title or abstracts. "MeSHLabeler" integrated these features into a learning to rank framework [11]. It outperformed the MTI of the day and got the best performance in 2014 BioASQ challenge Task A.

Yuqing Mao and Zhiyong Lu [12] proposed "MeSH Now" which obtains an initial list of MeSH candidates from similar documents found by k-NN. A learning to rank algorithm is used to rank these MeSH candidates, and some hand-crafted rules are used for post-processing and top-ranked MeSH selection.

The "MetaLabeler" proposed by Tsoumakas [13] addressed the multi-label classification problem as N binary classification problems [14] and solved them using linear Support Vector Machine(SVM), where N is the number of MeSH. The MeSH were ranked in terms of the SVM prediction score of each classifier. A regression model which is independent of the classification models was trained to predict K, the number of MeSH for each citation, and used it to select the top K MeSH in the ranked list.

The methods mentioned above successfully integrated machine learning model with the knowledge resource and achieved encouraging results. However, they have two kinds of shortcomings. Firstly, they can't represent the semantic of citations well by treating words as atomic symbols that ignore the words relation. Secondly, the machine learning methods currently used, such as SVM, k-NN and learning to rank model, need feature engineering in which researchers have to choose the features that fed to the machine learning model. The tedious feature engineering task not only requires domain knowledge but also lacks flexibility because some of the machine learning models lack interpretability in feature selection.

In this paper, we propose a deep learning model [15] with a serial multi-task learning structure to address the deficiency of the common methods for large-scale biomedical semantic indexing. We represent the citations as a sequence of word2vec [16] vectors. A bidirectional Gated Recurrent Unit (BGRU) [17] is used to take words order into consideration and generate the hidden representation of the citations. Finally, we design a serial multi-task structure [18] to get the model's output, which contains a primary multi-label classification task and a serial auxiliary regression task. The model outperforms the state-of-art MTI in F-measure and precision, and it achieves the highest precision among all the solutions in BioASQ Task 5A. Furthermore, the experiments show that the deep neural network with a serial multi-task paradigm converges significantly faster.

We also try an interesting experiment in which the semantic indexing task is seen as generating labels given the representation of a citation. We use Wasserstein Generative Adversarial Nets(WGAN) to address the label generating issue.

Methods

We proposed a deep serial multi-task learning model (SMTL) to solve biomedical semantic indexing problems. The outline of it is illustrated in Fig. 1.

We map the words in each citations to word2vec vectors that were pre-trained on 10,876,004 English abstracts in PubMed [19]. We use word embedding in consideration of the fact that it has overwhelming advantages [20] over other count based word representation methods.

We truncate or pad the input sequences to 360 words and feed the sequences to Bidirectional Recurrent Neural Network (BRNN) [21] to get the hidden representation. Gated Recurrent Units (GRU) [22] is used as the RNN cell. We stack three fully connected layers on top of the bidirectional GRU to perform the classification task. In order to alleviate the impact of the unbalanced data and to let multiple relevant tasks inform each other, we design an auxiliary regression task which adds up the elements of the output vector by the primary multi-label classification task. In addition, we use backpropagation algorithm to optimize the regression loss and the batch normalization [23] are adopted to speed up the training process.

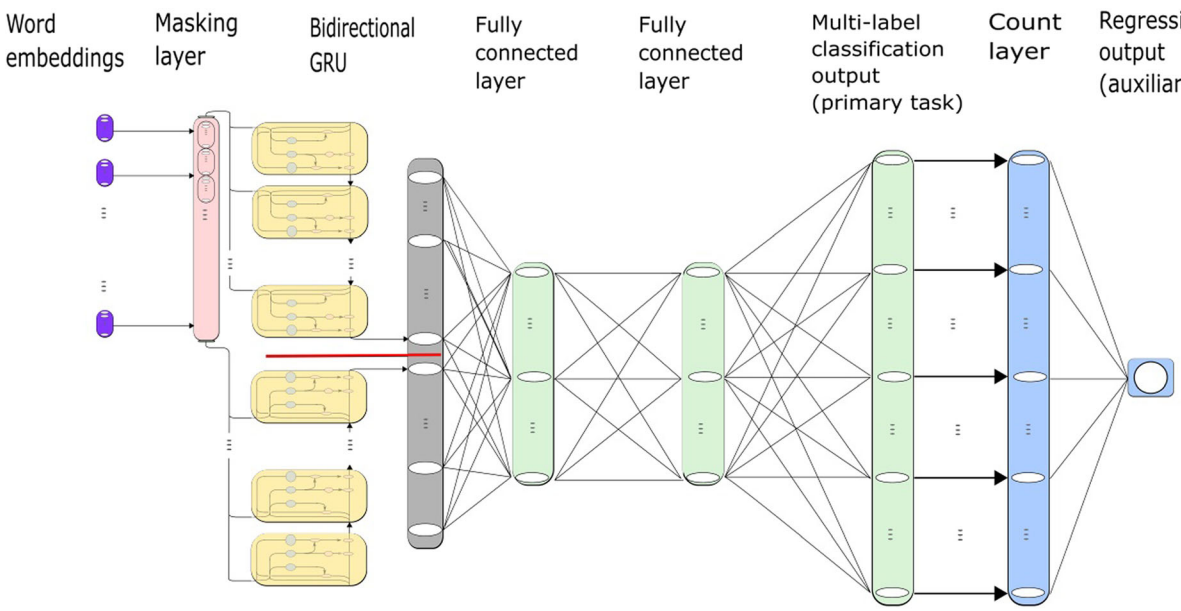

Fig. 1 Deep Serial Multi-Task Learning Model (SMTL). The purple part is the sequence of word embedding (**seq**) that represents the training citation. The pink part is the masking layer for unifying the length of each **seq**. The yellow part denotes the BGRU and the two directions of which are separated by a red line. The gray part is the output of the BGRU, which is the hidden state. The green part includes three fully connected layers and the rightmost one is the output layer of the primary classification. The blue part represents the structure of the SMTL, of which the rightmost one is the output layer of the auxiliary regression

Neural semantic word embedding

Word2vec word embedding can be seen as a predictive-based language model for the word and the context, which embedded the semantic information of the words. The most famous example of word2vec embedding is "vector("king") − vector("man") + vector("woman") ≈ vector("queen")" [24]. Baroni et al. [20] showed that this kind of neural semantic word embedding is superior to count-based distributional semantic models and other kinds of semantic representation in most of the Natural Language Processing tasks.Word2vec has two kinds of models and they are the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model [16].

The CBOW model makes use of both the previous and subsequent $n$words around the target word to predict the target word w_t. Conversely, the Skip-Gram model uses the center word to predict the surrounding words. They are shown in Fig. 2.

The objective function of CBOW is represented as Eq. 1.

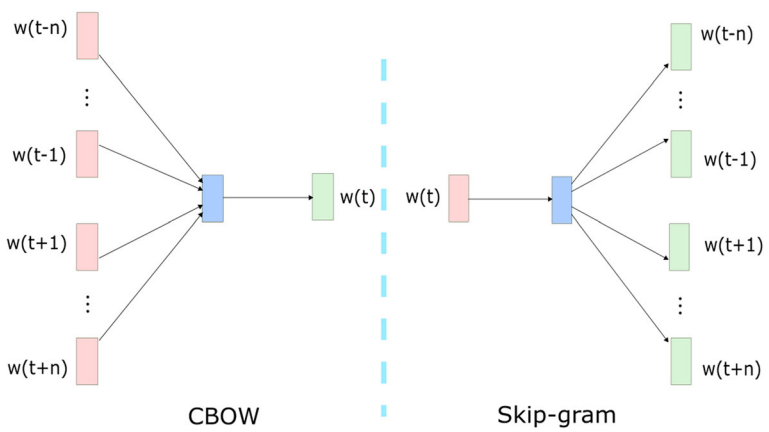

Fig. 2 The CBOW Model and the Skip-gram Model. The CBOW Model uses the previous n words w(t-1)…w(t-n) and the subsequent n words w(t + 1)… w(t + n) to predict the target word w(t). The Skip-gram Model uses the target word w(t) to predict the surrounding words w(t-n)…w(t-1), w(t + 1)…w(t + n)

$$J_\theta = \frac{1}{\mathrm{T}}\sum_{t=1}^{T} \log p(w_t|w_{t-n},\cdots,w_{t-1},w_{t+1},\cdots,w_{t+n}) \tag{1}$$

The objective function of Skip-Gram model is represented as Eq. 2.

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=-n}^{n} \log p\left(w_{t+j}|w_t\right) \quad ,(j\neq 0) \tag{2}$$

In order to get the word embedding, it is needed to maximize the objective function by maximizing the conditional probability. After the process of maximization, the network parameters corresponding to the words are the expected word embedding. We can simply implement the objective function using softmax function as illustrated in Eqs. 3 and 4, where $w_O$denotes the surrounding words, $w_I$denotes the center word, $\mathbf{v}$represent the input embedding, $\mathbf{v}'$represents the output embedding and Vrepresents the size of the vocabulary.

$$p(w_I|w_O) = \frac{\exp\left(\mathbf{v}'^{\mathrm{T}}_{w_t}\mathbf{v}_{w_I}\right)}{\sum_{w_i\in V}\exp\left(\mathbf{v}'^{\mathrm{T}}_{w_i}\mathbf{v}_{w_O}\right)} \tag{3}$$

$$p(w_O|w_I) = \frac{\exp\left(\mathbf{v}'^{\mathrm{T}}_{w_O}\mathbf{v}_{w_I}\right)}{\sum_{w=1}^{V}\exp\left(\mathbf{v}'^{\mathrm{T}}_{w}\mathbf{v}_{w_I}\right)} \tag{4}$$

However, softmax function is computationally complex since the denominator involves all the words in the vocabulary which could be huge in practice, thus word2-vec actually uses other more efficient methods instead of softmax function to represent the objective function.

In this paper, we use the pre-trained word embedding of 1,701,632 words provided by BioASQ which are trained on 10,876,004 English abstracts of biomedical articles from PubMed using skip-gram algorithm.

Bidirectional gated recurrent unit

Gated recurrent unit (GRU) is designed to resist gradient vanishing and exploding problems of the Recurrent Neural Network (RNN) and it has the ability to learn to forget or update the recurrent hidden state according to the context. The unit of GRU is illustrated in Fig. 3.

It can be seen that the GRU unit has an update gate and a reset gate. The update gate z_t^jdecides how much the unit update its content where $t$represents the time step t and j denotes the $j$thelement of the update gate vector $\mathbf{z}_t$. The reset gate r_t^jdecides how much the new candidate value $\tilde{h}_t^j$consider the previous hidden state $h_{t\text{-}1}^j$.

The update gate z_t^j is computed by Eq. 5.

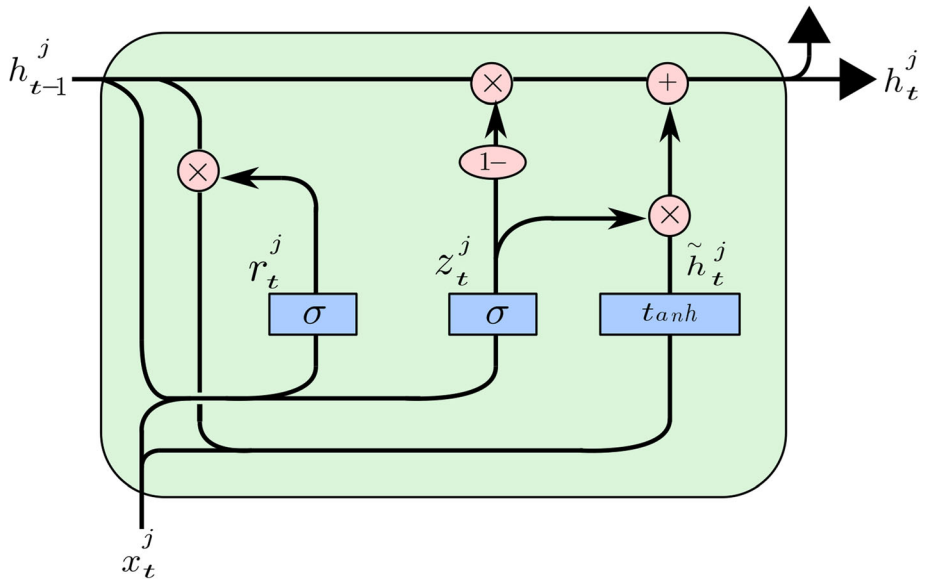

Fig. 3 The unit of GRU. $z_t^j$is an update gate, where the superscript$j$indicates the update gate is the $j$thelement in the update gate vector$\mathbf{z}_t$, the subscript $t$indicates the time step t. r_t^jdenotes a reset gate. h_{t-1}^j represents a hidden state of the time step t - 1, h_t^j represents a hidden state of the time step t and $\tilde{h}_t^j$is a candidate activation

$$z_t^j = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1})^j \quad (5)$$

where $\mathbf{W_Z}$ denotes the input weights matrix, $\mathbf{U_Z}$ denotes the recurrent weights matrix for the update gate, $\mathbf{x_t}$ is the input vector of the unit on the time step t and σdenotes the element-wise sigmoid function.

Similarly, the reset gate r_t^j is computed by Eq. 6.

$$r_t^j = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1})^j \quad (6)$$

where $\mathbf{W}_r$ denotes the input weights matrix, $\mathbf{U}_r$ denotes the recurrent weights matrix and $\mathbf{h}_{t-1}$ is the hidden state of the previous time step.

The activation h_t^j of the GRU is a linear interpolation between candidate activation $\tilde{h}_t^j$ and the previous activation h_{t-1}^j:

$$h_t^j = \left(1 - z_t^j\right)h_{t-1}^j + z_t^j\tilde{h}_t^j \quad (7)$$

The candidate update activation $\tilde{h}_t^j$ is computed by Eq. 8.

$$\tilde{h}_t^j = \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \bullet \mathbf{h}_{t-1}))^j \quad (8)$$

where $\mathbf{W}$ is the input weights and $\mathbf{U}$ is the recurrent weight matrix, $\bullet$ means elementwise multiplication.

The structure of GRU has the ability to capture dependencies over different time scales. In the units that learn to capture short-term dependencies, the reset gates tend to be active frequently. Meanwhile, in the units that capture long-term dependencies, the update gates tend to be more active. Most importantly, GRU can effectively alleviate the gradient vanishing and exploding problem which is the main disadvantage of standard RNNs. It makes the training process much easier.

Bidirectional recurrent neural network (BRNN) increases the amount of information available to the hidden representation of each time step allowing the recurrent units to use both the previous and the future information in the sequence. A bidirectional diagram is illustrated in Fig. 4.

Here, $\mathbf{H}_l$ denotes the output of the recurrent hidden units that propagate the information forward in time (from the left to the right) and $\mathbf{H}_r$ denotes the output of the recurrent hidden units that propagate the information backward in time(from the right to the left). Thus at each time step$t$, the output activations are computed by considering both the recurrent activation $\mathbf{H}_l^{(t-1)}$ of the previous time step and the recurrent activation $\mathbf{H}_r^{(t+1)}$ of the next time step. In this paper, we use GRU as the units of the BRNN, and concatenate $\mathbf{H}_l$ and $\mathbf{H}_r$ to get the hidden state $\mathbf{H}$.

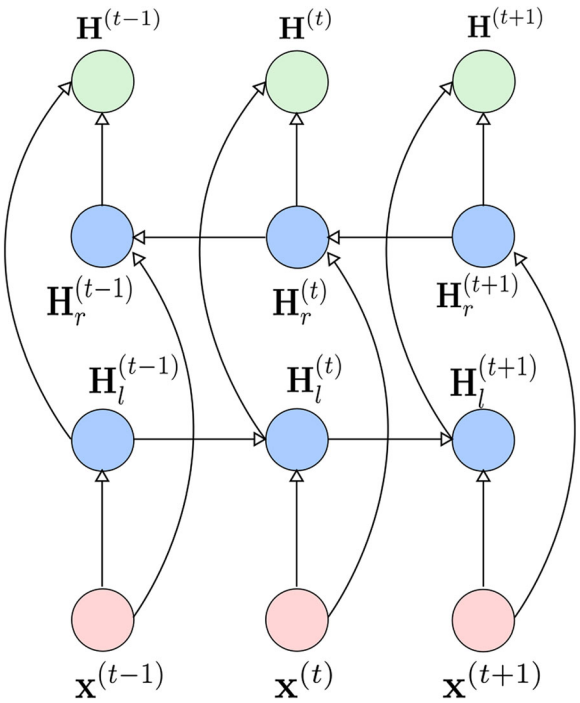

Fig. 4 The Structure of BRNN. The pink circles annotated with **x** denotes the input vectors, the blue circles annotated with $\mathbf{H}_l$ and $\mathbf{H}_r$ denote the output of the recurrent hidden units in two different directions, and the green circles annotated with **H** denotes the output of the BRNN, which is the hidden state vector generated by concatenating the recurrent hidden states from two directions

The bidirectional structure utilizes more information, thus strengthening the neural network's ability of representation. Some researchers have claimed that they got better performance in the research fields of machine translation [25], speech recognition [26] and so on. We also find that Bidirectional GRU has better performance than plain unidirectional GRU in our model.

Deep serial multi-task learning model

In multi-task learning (MTL) paradigm [18], more than one tasks are trained at the same time. The related tasks share part of the network structure, representation and the features extracted from each other so that these tasks can inform each other to learn better. Because the representation considers the need for all the tasks, the multi-task neural network tends to have a higher ability of generalization. Unrelated tasks can also benefit from the multi-task learning paradigm, too [27].

The loss of the MTL network is given by Eq. 9.

$$L_{MTL} = L_{pri} + \sum_{k=1}^{n}\lambda_k L_k \quad (9)$$

where L_{pri} refers to the loss of the primary task and L_k refers to the loss of the kth auxiliary task, λ_k is the

relative importance factor of the auxiliary task with respect to the primary task.

MTL has two main paradigms and the first one is the hard parameter sharing MTL paradigm, which shares the hidden layers among all the tasks and keeps several task-specific output layers as illustrated in Fig. 5.

The other one is the soft parameter sharing MTL paradigm [28], in which each task has its own relatively independent model and there are extra structures and parameters among these models for learning to share information.

Neural network based multi-label classification is naturally a multi-task learning process, in which the classification of each label can be seen as one task which shares the representation with the other classification tasks for other labels. The multi-label classification process in the neural network considers the relation between different labels, thus it has better performance over traditional machine learning methods that treat the multi-label classification as independent binary classifications.

As for the model we propose in this paper, we combine two related tasks in the novel serial multi-label learning structure. The primary task is the multi-label classification task, in which each label refers to a candidate MeSH. When a label is assigned the value of 1, it means that the corresponding MeSH should be assigned to the citation.

The structure of the serial multi-task paradigm is shown in Fig. 6. The count layer implements the operation described in Eq. 10. The primary classification task can be formalized as $P(\mathbf{seq}|\,\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the weights of the neural network and **seq** denotes the sequence of word embedding which is the input of the model.

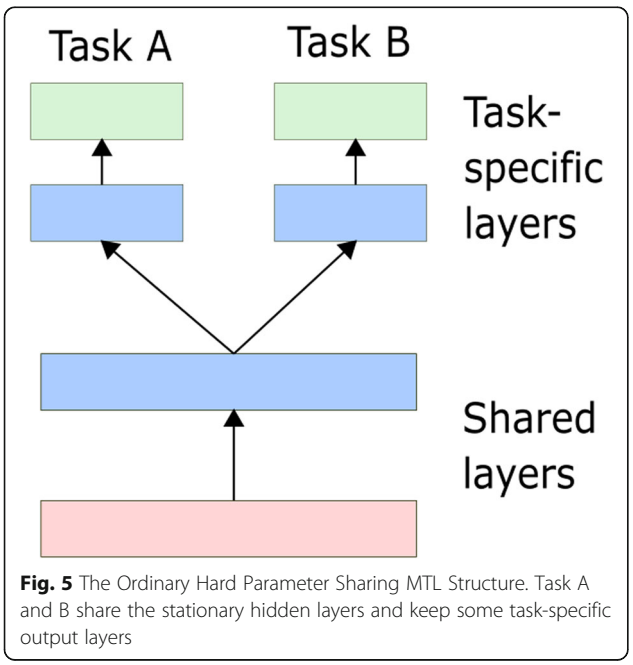

Fig. 5 The Ordinary Hard Parameter Sharing MTL Structure. Task A and B share the stationary hidden layers and keep some task-specific output layers

The auxiliary task is a regression task and we formalize it as $A(\mathbf{seq}|\,\boldsymbol{\theta})$. The calculation of the auxiliary task is shown in Eq. 10.

$$R = A(\mathbf{seq}|\boldsymbol{\theta}) = P(\mathbf{seq}|\boldsymbol{\theta}) \cdot \mathbf{v} \tag{10}$$

where R is a scalar which denotes the output of the auxiliary regression task, and it is the prediction for the total number of the labels (MeSH) that the citation has. Here, $\mathbf{v}$ is the vector $(1, 1, \cdots, 1)$ with 28,472 dimensions.

The operation demonstrated by Eq. 10 is equal to the process that counts the total number of the labels predicted by the primary classification task $P(\mathbf{seq}|\,\boldsymbol{\theta})$and uses it as the output of the auxiliary regression task $A(\mathbf{seq}|\,\boldsymbol{\theta})$.

During the optimization, the weights of the network will be updated to ensure the low loss of both the primary and auxiliary task. Thus this serial multi-task structure can deal with the primary task by knowing the total number of labels for the corresponding citations. The auxiliary task can inform the primary task and the generalization ability of the network is higher. We name the structure as the Serial Multi-Task Learning model (SMTL) for multi-label classification.

We use cross-entropy to measure the loss of the primary classification network, mean square error is used to measure the loss of the auxiliary regression network. Therefore, the loss function of SMTL is:

$$L = L_{pri} + L_{aux} \tag{11}$$

Here, $L_{pri}$is the cross-entropy loss of the primary classification network and $L_{aux}$is the mean square loss of the auxiliary regression network.

$$L_{pri} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{28472}\left(\begin{array}{l}\mathbf{y}_j^i \log P\left(\mathbf{seq}^i|\boldsymbol{\theta}\right)_j \\ +\left(1-\mathbf{y}_j^i\right)\log\left(1-P\left(\mathbf{seq}^i|\boldsymbol{\theta}\right)_j\right)\end{array}\right) \tag{12}$$

$$L_{aux} = \frac{\lambda}{N}\sum_{i=1}^{N}\left(\mathbf{z}^i - A\left(\mathbf{seq}^i|\boldsymbol{\theta}\right)\right)^2 \tag{13}$$

Here, **y**represents the ground true label of the primary classification task and **z** represents the label of the regression task, **seq**denotes the input of the model which is a word embedding sequence representing the training citation. The superscript i denotes the $i$ththe training sample of a selected mini-batch, and the subscript indicates the element in the corresponding vector.

The SMTL model makes use of the hard parameter sharing MTL paradigm, but it differs from the ordinary hard parameter sharing MTL model. The ordinary one's

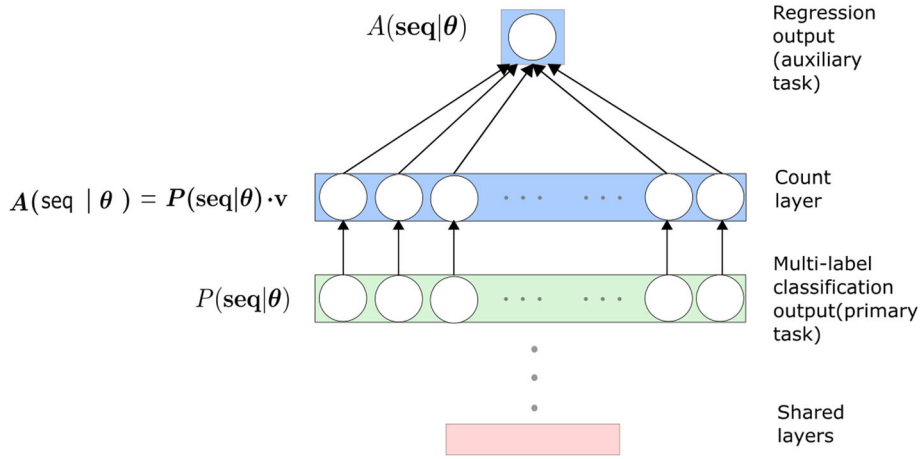

Fig. 6 The structure of the multi-task learning paradigm. The primary multi-label classification task is denoted as $P(\mathbf{seq}|\boldsymbol{\theta})$, the auxiliary regression task is denoted as $A(\mathbf{seq}|\boldsymbol{\theta})$and the relationship between *P*and *A*is described in eq. 10

tasks are parallel while the tasks in our model are serial and the auxiliary task is based on the primary task. It informs the primary task more directly and makes the network learn faster. SMTL shows high performance in the experiment.

The algorithm of SMTL is described in Algorithm 1.

Wasserstein Generative Adversarial Networks (WGAN) [29] is a promising generative model in which a generative model G captures the data distribution, and a discriminative model D estimates the probability that a sample came from the training data rather than G. We try to address the semantic indexing issue via

Algorithm 1. Serial Multi-Task Learning Algorithm

Symbol definition:

- $FC^k(\text{input\_dim}, \text{output\_dim})$ indicates the *k*th fully connected layer with *input_dim* input neuron and *output_dim* output neuron.
- **seq** denotes the input of the model which is a word embedding sequence representing the training citation.
- i denotes the *i*th training sample of a selected minibatch.

1. Make every training citation as a sequence of pre-trained word embedding.
2. Randomly initialize θ in the primary classification network $\mathrm{P}(\text{seq} \mid \theta)$.
3. **for** epoch = 1…M **do**
   - 3.1 *S*elect a random minibatch of N seq-label pairs (training samples).
   - 3.2 *T*runcate or pad every seq to the length of 360 words by using masking layer.
   - 3.3 **for** training sample from 1 to N **do**
     - *F*eed training sample to bidirectional GRU and get the output from two directions H_l and H_r.
     - *C*oncatenate H_l with H_r and get a hidden layer H with 540 dimensions.
     - *C*onnect 3 fully connected layers $FC^1(540, 540)$, $FC^2(540, 540)$, $FC^3(540, 28472)$ to H and get $\mathrm{P}(\text{seq}^i \mid \theta)$.
     - *C*alculate the output of the auxiliary regression: $A(\text{seq}^i \mid \theta) = P(\text{seq}^i \mid \theta) \cdot \mathrm{v}$

     end for
   - 3.4 *U*pdate θ by minimizing the loss: $L = L_{pri} + L_{aux}$

   where L_{pri} and L_{aux} is computed by equation 12 and 13 separately.

end for

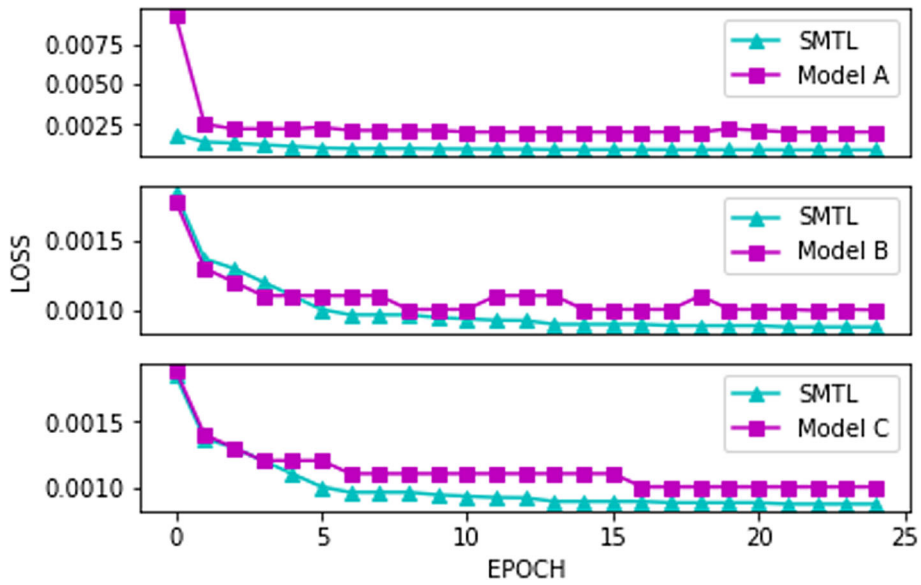

Fig. 7 The classification loss during the training process for different Model. The result shows that network with SMTL converges faster and has better performance than networks with an ordinary parallel multi-task learning structure (Model A) and network without a multi-task structure (Model B). It also shows that BGRU has better performance than unidirectional GRU (Model C) in SMTL

Wasserstein Generative Adversarial Nets (WGAN). The hidden state of the recurrent network in the trained SMTL model is a representation of the input citation, and we use it as the input of the generator to achieve a label combination. A discriminator is trained to distinguish the generated label combinations and the real label combinations. In the inference stage, we use the trained generator to predict the label combinations of the citations.

Results and discussion

Data set and details of the experiment

We use BioASQ Task 5A dataset [10] as training set, and the entire dataset contains 12,504,999 labeled citations annotated by NLM staff.

The 2017 BioASQ challenge Task A released the test data in many batches, and the participants' submitted solutions were evaluated by the performance on the batches. In order to compare our solution with

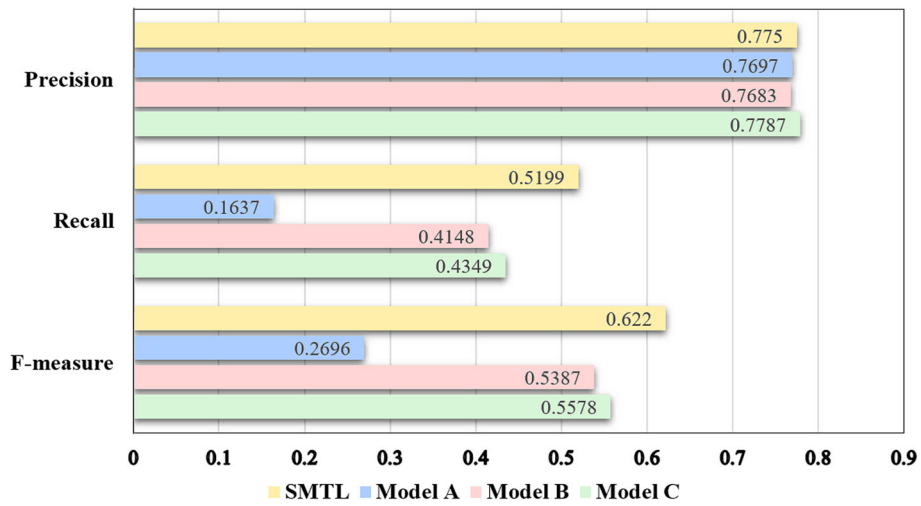

Fig. 8 Performance comparison of different models. SMTL performs better than Model A, B and C. The tightly coupled serial structure informs the tasks more so that the model spends less time learning the relationship between the primary and auxiliary tasks

Table 1 Performance comparison with different solutions in 2017 BioASQ task 5A

System	Precision	Recall	F-measure
SMTL	0.7750	0.5199	0.6220
DeepMeSH1	0.7052	0.6135	0.6561
auth3	0.6277	0.6366	0.6321
Default MTI	0.6408	0.6021	0.6209
MTI First Line Index	0.6624	0.5534	0.6030
iria-2	0.4853	0.5721	0.5251
Optimize Micro AUC	0.2890	0.2739	0.2812
Search system 1	0.2488	0.2907	0.2681

other participants' approaches sufficiently, we choose the last batch which is "week 5 batch 3" as the test data since more teams joined this evaluation than any other test batches.

With regard to the detail of the experiment, we use 200 dimensions word2vec embedding to represent the word. We use the masking layers to generate citations with variable length. The batch normalization layers are used to make the network less fragile during the training process and it also alleviates the overfitting problem. For the initialization of the GRU, we initialize the weights for the input vectors with Xavier uniform initializer [30]. The bias is initialized with zero vector. The hidden state of GRU is a vector of 270 dimensions, so that after the concatenation of the two directions, the hidden state is a vector of 540 dimensions. The two fully-connected layers are both of 540 dimensions. For the optimization process, we use Adam optimizer [31]. The deep learning library Theano [32] and Keras [33] are used to build our model. We use the first 3 million out of the total 12 million data in the Bioasq dataset as the training data and the model is trained on a Nvidia 1080ti GPU.

Experimental results

We design several other models and compare them with SMTL in the experiment.

SMTL: Our final model, which utilize a serial multi-task learning structure and bidirectional GRU.

Model A: An ordinary hard parameter sharing MTL model with Bidirectional GRU, and the MTL structure is demonstrated in Fig. 5 where Task A denotes the multi-label classification and Task B denotes the regression task.

Model B: A multi-label classification model with Bidirectional GRU.

Model C: A SMTL model which has a unidirectional GRU instead of a bidirectional GRU.

The binary cross-entropy loss of the multi-label classification task in different models during the training process is demonstrated in Fig. 7.

The main evaluation metrics in the BioASQ challenge Task A are Precision, Recall, and F-measure.

The performance evaluation results are demonstrated in Fig. 8.

The performance comparison with other solutions submitted by different participant teams in 2017 [34] is shown in Table 1 and Fig. 9. It's worth noticing that the solution "DeepMeSH1" [35] is the champion in the 2017 BioASQ Task 5A competition. Our experimental

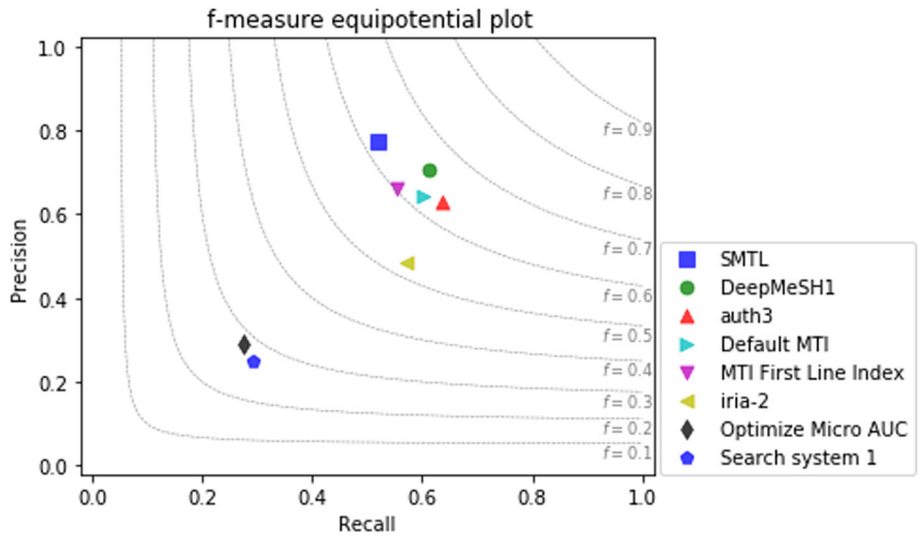

Fig. 9 F-measure equipotential plot of different solutions in 2017 BioASQ Task 5A. The best solutions from each team are exhibited in the Fig. 9. The plot shows that without any hand-craft feature or any sophisticated rule design, SMTL (the blue rectangle) achieves the highest precision among solutions of all the participants in 2017 BioASQ Task 5A

result labelled as "SMTL" achieves the highest precision among all the other solutions.

As demonstrated in Table 1 and Fig. 9, our SMTL outperforms the state-of-art solution MTI proposed by NLM on both precision and F-measure. As a reference, only the solutions from two participant teams [35, 36] beat MTI on F-measure in the 2017 BioASQ Task 5A challenge and neither of them adopted deep learning methods. Bidirectional GRU is good at capturing long term dependency in sequence data and does not need explicit feature extraction. SMTL structure increases the generalization ability of the model and makes the model converge faster in practice.

We use mean square error(MSE) as the metric for the auxiliary regression task, and the MSE loss on the test set is 27.54.

As depicted in Fig. 1, the recurrent neural network gives a representation of the input word sequences(citations). We use it as the input of the fully connected layers. And then the classification issue can also be viewed as training a generative model to generate a vector of 28,472 dimensions as the labels assigned to the citations.

We also try to use a WGAN framework to resolve the semantic indexing problem. The presentations in the trained SMTL are used as the prior vectors of G in WGAN to generate the 28,472 dimension label vectors. And 3 layers of fully connected neural networks are designed as a discriminative network to evaluate the authenticity of the generated vector.

The performance of the WGAN framework with the recurrent presentation is illustrated in Table 2 along with SMTL.

Since the model G and D in our WGAN framework are just fully connected neural networks with only 3 layers, it still remains great potential of the WGAN with the recurrent presentation.

Conclusions

We propose a novel deep serial multi-task learning model to address the issue of biomedical semantic indexing. The traditional methods ignore the relations among labels and need complicated feature engineering. Our model uses the word2vec word embedding to represent the words in the citations, and the Bidirectional GRU is used to create the representation of the data. This multi-task learning structure is different from an ordinary one because the auxiliary task originates directly from the primary task and the two tasks compose a serial structure. The regression task is motivated by dynamic threshold for classification task on unbalanced data. Without any handcrafted feature, our model outperforms the state-of-art baseline solution MTI in F-measure, and it has higher precision than the best solution "DeepMeSH1" in 2017 BioASQ Task 5A.

Furthermore, we are going to explore more auxiliary tasks to inform the multi-label classification task and apply the attention mechanism to our model for higher performance. And we will also pay more attention to investigate the possibilities of using WGAN for semantic indexing task.

Table 2 Performance comparison between WGAN framework and SMTL

System	Precision	Recall	F-measure
WGAN Framework	0.6376	0.5486	0.5898
SMTL	0.7750	0.5199	0.6220

Abbreviations

BGRU: Bidirectional Gated Recurrent Unit; BRNN: Bidirectional Recurrent Neural Network; CBOW: Continuous Bag-of-Words; GRU: Gated Recurrent Units; K-NN: K-Nearest Neighbor; MeSH: Medical Subject Headings; MTI: Medical Text Indexers; MTL: Multi-Task Learning; NLM: National Library of Medicine; RNN: Recurrent Neural Network; SMTL: Serial Multi-task Learning model; SVM: Support Vector Machine; UMLS: Unified Medical Language System; WGAN: Wasserstein Generative Adversarial Networks

Acknowledgement

Not applicable.

Funding

Publication charges were funded by National Science Technology Support Plan (Grant No. 2013BAH21B02–01). The research presented in this study was supported by the Natural Science Foundation of China (Grant No. 61375059, 61672065).

Availability of data and materials

The datasets supporting the conclusions of this article are available in http://participants-area.bioasq.org/.

About this supplement

This article has been published as part of *BMC Bioinformatics Volume 19 Supplement 20, 2018: Selected articles from the IEEE BIBM International Conference on Bioinformatics & Biomedicine (BIBM) 2017: bioinformatics*. The full contents of the supplement are available online at https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-19-supplement-20.

Authors' contributions

YD conceived the idea, gave important advice on the designing of the models and drafted the manuscript. YP proposed the ideas of the models, did the major coding work, and drafted the manuscript. CW implemented the idea of WGAN for multi-label text classification. JJ reviewed the manuscript and gave valuable advice on how to improve it. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Published: 21 December 2018

References

1. Lu Z, Kim W, Wilbur WJ. Evaluation of query expansion using MeSH in PubMed. Inf Retr. 2009;12(1):69–80.
2. Gu J, et al. Efficient semisupervised MEDLINE document clustering with MeSH-semantic and global-content constraints. IEEE Trans Cybern. 2013; 43(4):1265–76.
3. Richter RR, Austin TM. Using MeSH (medical subject headings) to enhance PubMed search strategies for evidence-based practice in physical therapy. Phys Ther. 2012;92(1):124–32.
4. https://www.nlm.nih.gov/mesh/meshhome.html. Accessed 3 July 2017.
5. https://www.nlm.nih.gov/bsd/bsd_key.html. Accessed 3 Jul 2017.
6. Liu K, et al. MeSHLabeler: improving the accuracy of large-scale MeSH indexing by integrating diverse evidence. Bioinformatics. 2015;31(12): 1339–47.
7. Aronson AR, et al. The NLM indexing initiative's medical text indexer. Medinfo. 2004;89:268-70.
8. Fung KW, Bodenreider O. Utilizing the UMLS for semantic mapping between terminologies. In: American medical informatics association (AMIA) annual symposium proceedings; 2005.
9. Aronson AR. Metamap: Mapping text to the umls metathesaurus. Bethesda: MD: NLM, NIH, DHHS; 2006. p. 1–26.
10. http://participants-area.bioasq.org/general_information/Task5a/. Accessed 14 Jul 2017.
11. Liu T-Y. Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval. 2009;3(3):225–331.
12. Mao Y, Lu Z. MeSH now: automatic MeSH indexing at PubMed scale via learning to rank. J Biomed semant. 2017;8(1):15.
13. Tsoumakas G, et al. Large-scale semantic indexing of biomedical publications at bioasq. In: BioASQ workshop; 2013.
14. Tsoumakas G, Katakis I. Multi-label classification: an overview. Int J Data Warehouse Min. 2006;3(3):1-13.
15. Du Y, Pan Y, Ji J. A novel serial deep multi-task learning model for large scale biomedical semantic indexing. In: 2017 IEEE international conference on bioinformatics and biomedicine (BIBM). Kansas: IEEE; 2017.
16. Mikolov T, et al. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems (NIPS); 2013.
17. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: MIT press; 2016.
18. Caruana R. Multitask learning, in Learning to learn. Berlin: Springer; 1998. p. 95–133.
19. https://www.ncbi.nlm.nih.gov/pubmed/. 14 Jul 2017.
20. Baroni M, Dinu G, Kruszewski G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: Association for Computational Linguistics (ACL); 2014.
21. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process. 1997;45(11):2673–81.
22. Cho K, et al. Learning phrase representations using RNN encoder--decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014.
23. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning (ICML); 2015.
24. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv. 2013;1301:3781.
25. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: International conference on learning representations (ICLR); 2015.
26. Graves A, Mohamed A-r, Hinton G. Speech recognition with deep recurrent neural networks. In IEEE international conference on acoustics, speech and signal processing (ICASSP). Vancouver: IEEE; 2013.
27. Paredes BR, et al. Exploiting unrelated tasks in multi-task learning. In: Artificial intelligence and statistics; 2012.
28. Duong L, et al. Low resource dependency parsing: cross-lingual parameter sharing in a neural network parser. In: Association for Computational Linguistics (ACL); 2015.
29. Arjovsky M, Chintala S, Bottou L. Wasserstein gan, in arXiv preprint. arXiv. 2017;1701:07875.
30. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics; 2010.
31. Kingma DP, Adam JB. A method for stochastic optimization. In: International conference for learning representations (ICLR); 2015.
32. http://deeplearning.net/software/theano/. 14 Jul 2017.
33. https://keras.io/. Accessed 1 Sep 2017.
34. http://participants-area.bioasq.org/results/5a/. Accessed 25 Sep 2017.
35. Peng S, et al. DeepMeSH: deep semantic representation for improving large-scale MeSH indexing. Bioinformatics. 2016;32(12):170–9.
36. Papanikolaou Y, et al. AUTH-Atypon at BioASQ 3: large-scale semantic indexing in biomedicine. In: Working notes for the conference and labs of the evaluation forum (CLEF); 2015.