Bioinformatics, 35(19), 2019, 3794–3802
doi: 10.1093/bioinformatics/btz142
Advance Access Publication Date: 9 March 2019
Original Paper

OXFORD

Data and text mining

MeSHProbeNet: a self-attentive probe net for MeSH indexing

Guangxu Xun[1,*], Kishlay Jha[1], Ye Yuan[2], Yaqing Wang[3] and Aidong Zhang[1]

[1]Department of Computer Science, University of Virginia, Charlottesville, VA 22904, USA, [2]Department of Information and Communication Engineering, Beijing University of Technology, Beijing 100022, China and [3]Department of Computer Science and Engineering, SUNY at Buffalo, Buffalo, NY 14260, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on November 15, 2018; revised on February 18, 2019; editorial decision on February 19, 2019; accepted on March 6, 2019

Abstract

Motivation: MEDLINE is the primary bibliographic database maintained by National Library of Medicine (NLM). MEDLINE citations are indexed with Medical Subject Headings (MeSH), which is a controlled vocabulary curated by the NLM experts. This greatly facilitates the applications of biomedical research and knowledge discovery. Currently, MeSH indexing is manually performed by human experts. To reduce the time and monetary cost associated with manual annotation, many automatic MeSH indexing systems have been proposed to assist manual annotation, including DeepMeSH and NLM's official model Medical Text Indexer (MTI). However, the existing models usually rely on the intermediate results of other models and suffer from efficiency issues. We propose an end-to-end framework, MeSHProbeNet (formerly named as xgx), which utilizes deep learning and self-attentive MeSH probes to index MeSH terms. Each MeSH probe enables the model to extract one specific aspect of biomedical knowledge from an input article, thus comprehensive biomedical information can be extracted with different MeSH probes and interpretability can be achieved at word level. MeSH terms are finally recommended with a unified classifier, making MeSHProbeNet both time efficient and space efficient.

Results: MeSHProbeNet won the first place in the latest batch of Task A in the 2018 BioASQ challenge. The result on the last test set of the challenge is reported in this paper. Compared with other state-of-the-art models, such as MTI and DeepMeSH, MeSHProbeNet achieves the highest scores in all the F-measures, including Example Based F-Measure, Macro F-Measure, Micro F-Measure, Hierarchical F-Measure and Lowest Common Ancestor F-measure. We also intuitively show how MeSHProbeNet is able to extract comprehensive biomedical knowledge from an input article.

Contact: gx5bt@virginia.edu

1 Introduction

MEDLINE (https://www.nlm.nih.gov/bsd/medline.html), the primary component of PubMed (https://www.ncbi.nlm.nih.gov/pubmed/), is a bibliographic database maintained by U.S. National Library of Medicine (NLM). As the online counterpart to MEDLARS (MEDical Literature Analysis and Retrieval System), MEDLINE currently covers more than 5200 worldwide journals, and contains more than 24 million references to journal articles in life sciences with a concentration on biomedicine. A distinctive feature of MEDLINE citations is that they are indexed with NLM Medical Subject Headings (MeSH) (https://www.nlm.nih.gov/mesh/meshhome.html). The MeSH thesaurus is a controlled vocabulary curated by the NLM experts and used for indexing, cataloging and searching for biomedical articles and information (Coordinators, 2016; Nelson *et al.*, 2004). Thus accurate MeSH indexing greatly facilitates biomedical research and knowledge discovery (Gopalakrishnan *et al.*, 2018; Jha *et al.*, 2017; Xun *et al.*, 2017c).

Currently, MeSH indexing for MEDLINE is mainly performed by the human experts in NLM. They have to go through the full text of each biomedical article to assign suitable MeSH terms. This ensures high accuracy of MeSH indexing but inevitably renders it very expensive. It is estimated that the average cost of annotating one biomedical article is around $9.4 (Mork *et al.*, 2013). More than 813 500 citations were added to MEDLINE in the year of 2017, and this number is rapidly increasing by the year. Apart from the huge monetary cost, manual MeSH indexing could also cause a possible delay before a newly published biomedical article gets annotated. This presents a challenge to the NLM experts to annotate biomedical articles efficiently and promptly.

Therefore, a system that can automatically annotate biomedical articles with relevant MeSH terms or assist human experts could be of great help. To this end, NLM has developed Medical Text Indexer (MTI) (Aronson *et al.*, 2004; Mork *et al.*, 2013, 2014). MTI takes the title and abstract of an article as the input and outputs relevant MeSH terms. MTI mainly consists of two modules: MetaMap Indexing (MMI) and PubMed-Related Citations (PRC). MetaMap (Aronson and Lang, 2010) is a software tool to extract biomedical concepts from the text. MMI recommends MeSH terms based on the biomedical concepts discovered by MetaMap. PRC recommends MeSH terms by looking at the MeSH annotations of similar citations in MEDLINE found by the PubMed-Related Articles (PRA) algorithm (Lin and Wilbur, 2007). The two sets of MeSH terms are combined to generate the final list of MeSH recommendations.

In order to continue to advance the development of MeSH indexing systems, the BioASQ challenge (http://bioasq.org/) on biomedical semantic indexing and question answering is held every year since 2013 (Tsatsaronis *et al.*, 2015). One of the two BioASQ tasks is to annotate new MEDLINE documents with relevant MeSH terms before MEDLINE curators annotate them manually. As new manual annotations become available, they are used to evaluate the performance of participating systems. Many new MeSH indexing systems have been proposed since then, e.g. MetaLabeler (Tang *et al.*, 2009), MeSHLabeler (Liu *et al.*, 2015) and DeepMeSH (Peng *et al.*, 2016). MetaLabeler trains an independent binary classifier for each MeSH term; MeSHLabeler proposes to integrate MetaLabeler with multiple evidence such as similar publications and term frequencies; and DeepMeSH is an improved version of MeSHLabeler by incorporating deep semantics in the word embedding space (Mikolov *et al.*, 2013; Yuan *et al.*, 2017, 2018). They also have another classifier to determine the number of MeSH terms to recommend.

Formally speaking, MeSH indexing is a multi-label classification task, where each MeSH term can be regarded as a class label and each article can be labeled with multiple MeSH terms. Compared with regular multi-label classification problems, the large size of MeSH vocabulary and the imbalanced nature of different MeSH terms pose more challenges to the MeSH indexing problem. Currently there are more than 28 000 distinct MeSH terms and new MeSH terms are added to the vocabulary every year. The most frequent MeSH term 'humans' appears around 8 000 000 times in MEDLINE citations, while there are hundreds of infrequent terms that appear less than 10 times. These challenges have been taken into consideration by the previous researchers when designing their MeSH indexing systems. However, there are some other challenges and limitations that previous systems seem to have overlooked. First, the biomedical articles are sequences in nature, but most previous systems are based on models that cannot be easily used for sequential modeling in an end-to-end fashion, such as K-Nearest-Neighbors (KNN) and Support Vector Machine (SVM). Second, most previous systems train independent classifiers for each MeSH term, resulting in extremely long training time, high disk usage and inability to collaboratively train the classifier and exploit the correlation between different MeSH terms at the same time. Third, every time a new biomedical article is added, the previous MeSH indexing systems need to find similar articles from the MEDLINE database. In other words, millions of MEDLINE articles have to be stored with the system and a thorough search has to be done for each indexing. This further exacerbates the time and space consumption for the existing systems.

Deep learning is a family of machine learning methods that employ multiple processing layers to learn representations of data with multiple levels of abstraction (LeCun *et al.*, 2015). Attention mechanism (Bahdanau *et al.*, 2014; Vaswani *et al.*, 2017) including self-attention (Lin *et al.*, 2017) enables deep learning models to selectively pay attention to different parts of the input and provides interpretability. Deep learning and attention mechanism have improved the state-of-the-art in many research fields such as machine translation (Bahdanau *et al.*, 2014) and text classification (Lin *et al.*, 2017).

Inspired by the aforementioned challenges and the rapid development of deep learning techniques, we propose an end-to-end deep framework for this multi-label classification task. We propose to train a unified classifier instead of a large number of independent classifiers, thus the efficiency is improved and the correlation between different MeSH terms can be learned simultaneously. More specifically, the new framework is a self-attentive deep neural network classifier. The proposed model contains three major components: a bidirectional Recurrent Neural Network (RNN), a number of self-attentive MeSH probes and a multi-view neural classifier. The proposed model is able to extract different aspects of biomedical knowledge from an input article. RNNs are naturally suitable for sequential text data, and by mapping the input text into the embedding space, RNNs can benefit from word embeddings that carry semantic regularities (Bengio *et al.*, 2006; Mikolov *et al.*, 2013; Xun *et al.*, 2017a, b). By feeding RNN hidden states to self-attentive MeSH probes, each article can be converted into a fixed-dimension feature matrix. The multi-view neural classifier is a unified multi-label classifier that considers the extracted feature from the input text, the journal information as well as the correlation between different MeSH terms. The new framework is named MeSHProbeNet (in the 2018 BioASQ challenge, we used the name xgx for our system). To sum up, MeSHProbeNet has the following advantages:

- MeSHProbeNet is an end-to-end framework that does not rely on any other existing MeSH indexing systems or software tools.
- MeSHProbeNet is a unified multi-label classifier, thus very efficient in terms of training time consumption and disk usage for this large-scale MeSH indexing task.
- The bidirectional RNN of MeSHProbeNet is able to make use of the word embedding semantics and capture the context-dependent information via sequence modeling.
- The MeSH probes on top of the RNN allow us to extract different aspects of biomedical knowledge from the input article and represent it as a fixed-dimension feature matrix.
- The multi-view classifier considers both the extracted features and the journal information.
- MeSHProbeNet, as a unified multi-label classifier, simultaneously exploits the correlation between different MeSH terms as it is being trained.

The efficacy of MeSHProbeNet was demonstrated in Task A of the 2018 BioASQ challenge. We also provide an interpretability visualization of the MeSH probes to show how the proposed model

selectively pays attention to different parts of the input article and how different aspects of biomedical knowledge are extracted by the MeSH probes. We also perform an ablation study of MeSHProbe to show the importance of MeSH probes.

2 Methodology

The overview of our proposed MeSHProbeNet model is shown in Figure 1. MeSHProbeNet is a self-attentive deep neural network, which is able to predict a set of MeSH terms for a biomedical article based on its textual content and journal information. The textual content of a biomedical article includes the title, abstract and body (in the challenge dataset, only the title and abstract are available). The journal information refers to the name of the journal it was published in.

Briefly speaking, MeSHProbeNet consists of three main components. The first component is a bidirectional RNN on the textual contents of biomedical articles. The second component is a set of self-attentive MeSH probes, which are responsible for extracting useful information from the RNN hidden states and converting articles of various lengths into fixed-dimension feature matrices. The third component is a multi-view neural classifier which combines the extracted textual information with the journal information, and generates a set of relevant MeSH terms.

We will introduce our model according to how to convert the textual contents into fixed-dimension matrices and how to recommend MeSH terms based on the combined information.

2.1 Bidirectional RNN

The bidirectional RNN reads the textual contents of a biomedical article, i.e. the concatenation of the title and the abstract, and generates a hidden state for each word in the textual contents, as shown in the bottom left part of Figure 1. RNNs model texts in a sequential fashion and are able to capture the dependency between adjacent words. Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho *et al.*, 2014) have proven to be more effective in modeling long sequences than the vanilla RNN (Chung *et al.*, 2014). In MeSHProbeNet, we use a bidirectional GRU, as GRUs are simpler and perform on par with LSTMs. Suppose we have a sequential text which has T words as the input, i.e. the concatenation of the title and the abstract in our case. The first step is to represent the text as a sequence of T word embeddings:

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\},$$

where $\boldsymbol{x}_t$ is a D_w dimensional real-valued vector, denoting the embedding for the tth word in the input article. Thus a biomedical article can be represented as a T-by-D_w matrix, which is the concatenation of all the word embeddings in it. Then we feed article embedding matrix $\boldsymbol{X}$ to the bidirectional GRU:

$$\overrightarrow{\boldsymbol{h}_t} = \overrightarrow{GRU}(\boldsymbol{x}_t, \overrightarrow{\boldsymbol{h}_{t-1}}),$$
$$\overleftarrow{\boldsymbol{h}_t} = \overleftarrow{GRU}(\boldsymbol{x}_t, \overleftarrow{\boldsymbol{h}_{t+1}}),$$

where $\overrightarrow{\boldsymbol{h}_t}$ and $\overleftarrow{\boldsymbol{h}_t}$ are two U dimensional real-valued vectors, standing for the hidden states for the tth word in normal direction and reverse direction, respectively. By concatenating $\overrightarrow{\boldsymbol{h}_t}$ and $\overleftarrow{\boldsymbol{h}_t}$, we derive a $2U$ dimensional hidden state $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}_t}, \overleftarrow{\boldsymbol{h}_t}]$ which includes both the normal direction sequential information and the reverse direction sequential information at time stamp t. Hence, the hidden states of the input article can be represented as a T-by-$2U$ matrix:

$$\boldsymbol{H} = [\boldsymbol{h}_1; \boldsymbol{h}_2; \ldots; \boldsymbol{h}_t; \ldots; \boldsymbol{h}_T].$$

2.2 Self-attentive MeSH probes

One simple way to obtain the summary of the input article is to use the last hidden states of the bidirectional GRU: $[\overrightarrow{\boldsymbol{h}_t}; \overleftarrow{\boldsymbol{h}_1}]$. Although GRUs have proven to be more effective at modeling long sequences than the vanilla RNNs, their performances on really long sequences are still limited, such as the entire title and abstract text in our case. Hence, we propose to use a self-attentive MeSH probe mechanism to extract comprehensive aspects of biomedical information from the input article. Each MeSH probe carries one aspect of biomedical knowledge, and only

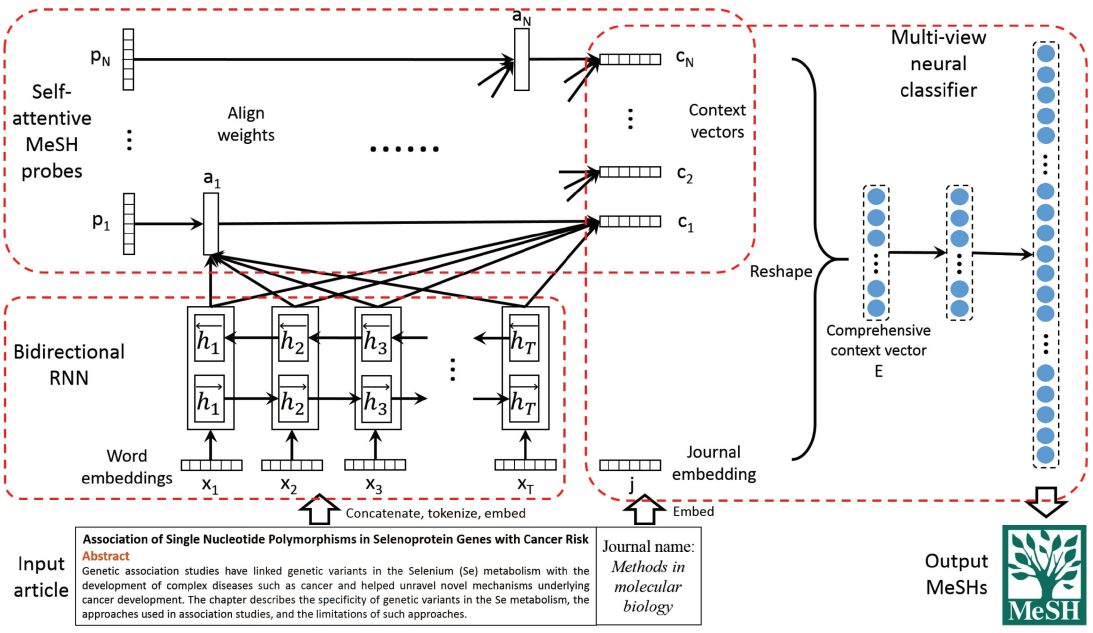

Fig. 1. The framework of MeSHProbeNet

pays attention to the RNN hidden states that contain related information. For instance, a MeSH probe that carries disease related knowledge is able to selectively extract the RNN hidden states that are related to disease. Specifically, one MeSH probe generates a weight vector for the RNN hidden states and multiply the RNN hidden states with the weight vector. Therefore, the resulting weighted RNN hidden state can be regarded as a summation of the input biomedical article with respect to the biomedical knowledge carried by the MeSH probe. With the help of the MeSH probe, biomedical articles of different lengths can be represented as a fixed-length vector containing related information. In fact, we can have multiple MeSH probes to cover multiple aspects of biomedical knowledge. Hence, given a certain number of MeSH probes, we can obtain a fixed-dimension output matrix that carries corresponding biomedical knowledge extracted from the input article.

More specifically, a MeSH probe is an inherent vector of MeSHProbeNet, which is associated with one specific aspect of biomedical knowledge. As with the GRU hidden state, the dimension of a MeSH probe is also $2U$. The goal of a MeSH probe is to extract related biomedical information from the input article and output a fixed-length vector. We achieve that by calculating a weighted combination of the T GRU hidden states. In particular, given MeSH probe $\boldsymbol{p}_n$, we first take all the GRU hidden states $\boldsymbol{H}$ as the input and then compute a normalized weight vector $\boldsymbol{\alpha}_n$:

$$\boldsymbol{\alpha}_n = \text{softmax}(\boldsymbol{p}_n \boldsymbol{H}^T).$$

Hence, $\boldsymbol{\alpha}_n$ is a 1-by-T vector where element α_{nt} indicates the weight for the tth GRU hidden state and all the weights sum up to 1:

$$\alpha_{nt} = \frac{\exp(\boldsymbol{p}_n \cdot \boldsymbol{h}_t)}{\sum_{t'=1}^{T} \exp(\boldsymbol{p}_n \cdot \boldsymbol{h}_{t'})}.$$

By taking the inner product between MeSH probe $\boldsymbol{p}_n$ and each GRU hidden state, MeSH probe $\boldsymbol{p}_n$ assigns higher weights and pays more attention to the hidden states that carry related biomedical knowledge. Then we can use the weighted summation of the GRU hidden states according to the weights in $\boldsymbol{\alpha}_n$ to represent the input article, denoted as context vector $\boldsymbol{c}_n$,:

$$\boldsymbol{c}_n = \boldsymbol{\alpha}_n \boldsymbol{H} = \sum_{t=1}^{T} \alpha_{nt} \cdot \boldsymbol{h}_t.$$

Context vector $\boldsymbol{c}_n$ is a $2U$ dimensional vector, which pays attention only to the parts of the input article related to MeSH probe $\boldsymbol{p}_n$. However, for a research article, one MeSH probe is normally insufficient as there are multiple aspects in it. For example, a research article about Alzheimer's disease is probably also related to aging and treatments. Therefore, to get a more comprehensive representation of the input article, we need multiple MeSH probes to pay attention to different aspects of the article, for instance, one probe for disease, one probe for treatments, another probe for anatomy, and so on. As illustrated by the top left part of Figure 1, if we want to examine N different aspects of the input article, N MeSH probes are required:

$$\boldsymbol{P} = [\boldsymbol{p}_1; \boldsymbol{p}_2; \ldots; \boldsymbol{p}_N],$$

where $\boldsymbol{P}$ is a N-by-$2U$ matrix composed of N different MeSH probes. Accordingly, we can obtain a N-by-T weight matrix A, where each row $\boldsymbol{\alpha}_n$ denotes the weight vector with respect to each MeSH probe $\boldsymbol{p}_n$:

$$\boldsymbol{A} = \text{softmax}(\boldsymbol{P}\boldsymbol{H}^T),$$

where the softmax function is performed along the second dimension of the input. Hence, with the help of multiple MeSH probes, we are able to extract different aspects of biomedical knowledge from the input article, and represent it with a N-by-$2U$ context matrix $\boldsymbol{C}$:

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{H}.$$

2.3 Multi-view neural classifier

With the help of the bidirectional RNN and the MeSH probes, now we are able to convert a biomedical article of arbitrary length to a fixed-dimension context matrix, where each row represents one particular aspect of the input article. In fact, for each input article, we also have its journal information in addition to the textual content. This journal information is quite useful, as biomedical journals typically have a definite research topic and focus on a specific research domain. Therefore, it is natural to expect that research papers published in the same journal tend to be annotated with MeSH terms related to the journal's research focus. To take the journal information into consideration, our multi-view neural classifier has a journal embedding module, where each journal name can be converted to a unique vector of length D_j. Thus, by reshaping the extracted context matrix $\boldsymbol{C}$ to a vector and concatenating it with the journal embedding, we are able to obtain a context vector of length $N * 2U + D_j$ that carries all the available information of the input article: the title, the abstract and the journal information. We denote this comprehensive context vector by $\boldsymbol{E}$.

Our task is to annotate a biomedical article with suitable MeSH terms. Hence, having extracted comprehensive context vector $\boldsymbol{E}$ from the input article, what we need to do next is to learn a function f that maps context vector $\boldsymbol{E}$ to V conditional probability distributions, where V is the size of the MeSH vocabulary. The output of f is a vector whose ith element estimates the probability that the ith MeSH term should be assigned to the current article:

$$P(m_i = 1|\boldsymbol{E}) = f(i, \boldsymbol{E}),$$

where m_i denotes the ith MeSH term in the MeSH vocabulary. Function f could be implemented by a feed forward neural network. We employ a three layer neural network, whose first layer is the input context vector $\boldsymbol{E}$, second layer is the hidden layer with ReLU activation and third layer is the output layer. More precisely, the multi-layer neural network calculates the following function, with a sigmoid output layer to guarantee each output neuron being a probability in the range of [0, 1]:

$$f(\boldsymbol{E}) = \sigma(\boldsymbol{W}_2 \text{ReLU}(\boldsymbol{W}_1 \boldsymbol{E} + \boldsymbol{b}_1) + \boldsymbol{b}_2), \quad (1)$$

where $\sigma(\cdot)$ is the element-wise sigmoid function, $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the weight matrices for each layer, and $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ are the biases. During training, each biomedical article comes with several manually annotated MeSH terms. So it can be regarded as a multi-label classification task, where the ground truth label is a V-length binary vector whose ith element is set to 1 if the ith MeSH term is assigned to the current article and set to 0 otherwise. We represent this ground truth vector by g. Therefore, given a biomedical article k, the objective is to minimize the following binary cross entropy loss:

$$L_k = -\sum_{i=1}^{V} (g[i] \cdot \log(f(i, \boldsymbol{E})) + (1 - g[i]) \cdot \log(1 - f(i, \boldsymbol{E}))).$$

Let K be the total number of articles in the training dataset, then the overall training objective is:

$$L = \sum_{k=1}^{K} L_k. \quad (2)$$

Note that unlike most previous works that train a binary classifier for each MeSH term separately, we train a unified multi-label classifier that considers all the MeSH terms simultaneously. The advantages of training a unified multi-label classifier are manifold. First, the efficiency for both training and predicting can be drastically improved by learning a unified classifier as there are more than 28 000 distinct MeSH terms. Second, by learning a unified classifier, the semantics of the word embeddings and journal embeddings can be shared by all MeSH terms. Third, the correlation between different MeSH terms is automatically exploited and carried by neural network weights $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$. If one MeSH term frequently co-occurs with other MeSH terms, for example, 'Alzheimer disease' is often accompanied by 'aged, 80 and over', this co-occurrence will influence the corresponding neurons in $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ simultaneously, and thus the correlation and dependency relationship can be captured.

Infrequent MeSH terms also benefit from this unified architecture. Hundreds of infrequent terms appear in less than 10 articles. Therefore, if an independent classifier is trained for each infrequent term, the classifier inevitably suffers from the lack of training data and would encounter tons of out-of-vocabulary words during prediction. By sharing parameters across all MeSH terms, such as word embeddings and weight matrices, the unified classifier is able to tackle the problem of lacking training data and the out-of-vocabulary problem for infrequent MeSH terms. In addition, infrequent terms can further take advantages of the correlation information in the unified classifier, especially if an infrequent term always co-occurs with some specific frequent terms.

The free parameters of the whole model are the word embeddings, the GRU weight matrix, the GRU bias, the MeSH probes, the journal embeddings, the fully connected neural network weight matrices and biases. Let θ denote the overall free parameter set. Then training can be achieved by looking for θ that minimizes the training corpus binary cross entropy loss in Eq. 2 via stochastic gradient descent. Stochastic gradient descent iteratively updates the free parameters after feeding the kth article of the training corpus:

$$\theta \leftarrow \theta - \eta \frac{\partial L_k}{\partial \theta},$$

where η is the learning rate.

In the prediction phase, there are two approaches to determine the final MeSH terms based on the output of function f in Eq. 1. One approach is to find the optimal thresholds for each MeSH term on a held-out validation set. The other approach is to learn another neural network to predict the number of related MeSH terms given a biomedical article. In practice, we adopt the first approach in the prediction phase, as it is more efficient and intuitive.

3 Experiments

We carry out experiments on the large-scale MeSH indexing task to demonstrate the efficacy of our MeSHProbeNet model. To illustrate how MeSHProbeNet extracts different aspects of biomedical knowledge from the input articles, we visualize MeSH probes and their attentions on different parts of the input sequence. To investigate the quality of the MeSH terms recommended by MeSHProbeNet, we participated in the 2018 BioASQ challenge and compare its performance with several state-of-the-art MeSH indexing systems, including MTI and DeepMeSH. Our system won the first place in the third batch of the challenge.

3.1 Dataset and experimental settings

The training dataset is downloaded from the challenge webpage (http://participants-area.bioasq.org/general_information/Task6a/). It contains 13 486 072 biomedical articles which are annotated with relevant MeSH terms by the PubMed human experts. On average, 12.69 MeSH terms are assigned to each article. In total, 28 340 distinct MeSH terms are covered by the training dataset. For each article in the training dataset, we have the unique identifier of the article (PMID), the title of the article, the abstract of the article, the year the article was published, the journal the article was published in and a set of MeSH terms assigned to the article.

In the preprocessing step, all non-alphanumeric characters, stop words and words with a total frequency lower than 10 are removed, and all words are converted to lowercase. The dimensionalities of word embeddings and journal embeddings are set to 250 and 100, respectively. The number of GRU layers is set to 2. The size of the GRU hidden unit is set to 200 per direction, thus 400 for a bidirectional unit. The dimensionality of MeSH probes is also set to 400 accordingly. The number of different MeSH probes that the model contains is 25. The multi-view neural classifier has a hidden layer of 10 000 units. We deploy 0.5 dropout, 0.00001 L2 regularization and snapshot ensemble (Huang *et al.*, 2017) to prevent over-fitting. The learning rate for stochastic gradient descent is set to 0.0005 and we also clip the gradients whose values are larger than 5.

3.2 MeSH probe visualization

Interpretability is one of the advantages of MeSHProbeNet. For the users of automatic MeSH indexing models, a good model should not only be accurate, but also be able to tell them which parts of the input support the recommended MeSH terms. For instance, the human indexers can achieve higher annotation efficiency with the help of interpretable MeSH indexing models, as this interpretability of automatic MeSH indexing models can provide them with evidence for adding or deleting a recommended MeSH term.

The interpretability of MeSHProbeNet can be achieved through examining the attention weight matrix $\boldsymbol{A}$. Each row $\boldsymbol{a}_n$ in attention weight matrix $\boldsymbol{A}$ represents the weight vector with respect to MeSH probe $\boldsymbol{p}_n$. Each element in weight vector $\boldsymbol{a}_n$ corresponds to how much attention MeSH probe $\boldsymbol{p}_n$ pays to each GRU hidden state and each word. Thus we can visualize the attention by drawing a heat map of the weight vector.

It is worth mentioning that another advantage of MeSHProbeNet is its unsupervised nature: the MeSH probes are learned in a completely unsupervised fashion. The training objective function drives the MeSH probes to extract comprehensive aspects of biomedical knowledge with each probe focusing on one specific aspect. In other words, we do not need any prior knowledge, external knowledge or human guidance for the MeSH probes. The probes are automatically learned and are able to capture biomedical semantics during training and provide interpretability.

We select two articles from the last test set of the 2018 BioASQ challenge, whose PMIDs are '29439706' and '27130306', to visualize MeSH probes and show the interpretability in Figure 2. For article 29439706, the ground truth MeSH terms assigned by human curators are 'biomedical research', 'disease eradication', 'HIV infections', 'humans', 'public health' and 'terminology as topic'; and the MeSH terms assigned by MeSHProbeNet are 'humans', 'HIV infections', 'research', 'disease eradication', 'public health', 'AIDS vaccines', 'HIV-1' and 'anti-HIV agents'. For article 27130306, the ground truth MeSH terms assigned by human curators are 'Alzheimer disease', 'Bayes theorem', 'Europe', 'humans', 'incidence' and 'prevalence'; and the MeSH terms assigned by MeSHProbeNet are 'prevalence', 'humans', 'male', 'female', 'Alzheimer disease',

Fig. 2. MeSH probe interpretability visualization. (**a**) MeSH probe No.2 extracts disease related information from article 29439706. (**b**) MeSH probe No.2 extracts disease related information from article 27130306. (**c**) MeSH probe No.11 extracts Alzheimer's related information from article 27130306

'aged', 'aged, 80 and over', 'incidence', 'Bayes theorem' and 'Europe'.

We first demonstrate how MeSH probe No.2 extracts disease related information from different articles in Figure 2a and b. The values below each word denote the normalized weights. We can see that MeSH probe No.2 pays more attention to words like 'HIV', 'virus' and 'disease'. Some words such as 'incidence' and 'background' also have high attention weights. This is because of the sequential nature of RNNs and the system recognizes those words as related words in the context of 'disease'. Then in Figure 2b and c, we demonstrate how two different MeSH probes extract two different aspects of biomedical knowledge from the same article. As we just mentioned, in Figure 2b MeSH probe No.2 extracts disease related information. While in Figure 2c, MeSH probe No.11 extracts Alzheimer's related information. One can observe that in this article, MeSH probe No.2 is sensitive to words like 'disease' and 'epidemiology', while MeSH probe No.11 is sensitive to words like 'Alzheimer's' and 'elderly'.

3.3 Evaluation metrics

In order to evaluate MeSH indexing performance, two sets of measures are used, one flat and one hierarchical.

The flat measures consist of accuracy and three sets of F-measure based metrics: Example Based F-Measure (EBF), Macro F-Measure (MaF) and Micro F-Measure (MiF). Accuracy represents the fraction of correct predictions. EBF is computed in a per data point manner. For each predicted label, only its score is computed, and then these scores are aggregated over all the data points. EBF for each data point can be computed as the harmonic mean of standard precision (EBP) and recall (EBR) for each data point. MaF, Macro Precision (MaP) and Macro Recall (MaR) give equal weight to each MeSH class. Frequent MeSH terms and infrequent MeSH terms are equally important. Thus MaP and MaR are calculated as the average precision and recall over all the MeSH classes. MiF, Micro Precision (MiP) and Micro Recall (MiR) aggregate the contributions of all MeSH classes to compute the average metric. Frequent MeSH terms therefore have higher weights than infrequent MeSH terms. We can see that different F-Measures have different focus, for example, MiF focuses more on the frequent MeSH terms, while MaF treats all MeSH terms equally regardless of their frequencies. Since the BioASQ challenge evaluates the systems based on MiF, we will also take MiF as our major measure.

The MeSH vocabulary is organized in a hierarchical structure. Thus hierarchical measures are also used to evaluate the performance, including Hierarchical Precision (HiP), Hierarchical Recall (HiR), Hierarchical F-Measure (HiF), Lowest Common Ancestor Precision (LCA-P), Lowest Common Ancestor Recall (LCA-R) and Lowest Common Ancestor F-measure (LCA-F) (Kosmopoulos *et al.*, 2015).

Table 1. Comparison results based on the flat measures

Models	MiP	MiR	MiF	EBP	EBR	EBF	MaP	MaR	MaF	Acc
Access Inn MAIstro	0.2351	0.3423	0.2788	0.2488	0.3558	0.2775	0.3942	0.4641	0.3905	0.1669
MeSHmallow	0.3798	0.2707	0.3161	0.3798	0.2661	0.3042	0.1333	0.0049	0.0037	0.1915
UMass Amherst T2T	0.5239	0.4759	0.4988	0.5408	0.4789	0.4881	0.4179	0.2526	0.2481	0.3392
iria	0.4654	0.5792	0.5161	0.4609	0.5929	0.5058	0.4271	0.4658	0.4147	0.3525
MTIFL	0.6730	0.5977	0.6332	0.6833	0.6121	0.6264	0.6377	0.5622	0.5408	0.4759
MTI	0.6475	0.6473	0.6474	0.6540	0.6648	0.6418	0.6086	**0.6084**	0.5667	0.4911
AttentionMeSH	0.6833	0.6447	0.6635	0.6853	0.6488	0.6497	0.6178	0.4943	0.4827	0.4982
DeepMeSH	0.6761	0.6517	0.6637	0.6767	0.6659	0.6544	0.6352	0.5455	0.5281	0.5020
MeSHProbeNet	**0.7172**	**0.6611**	**0.6880**	**0.7193**	**0.6736**	**0.6789**	**0.6782**	0.5804	**0.5671**	**0.5310**

Table 2. Comparison results based on the hierarchical measures

Models	LCA-P	LCA-R	LCA-F	HiP	HiR	HiF
Access Inn MAIstro	0.2722	0.3615	0.2964	0.4696	0.5921	0.5043
MeSHmallow	0.4000	0.2369	0.2871	0.5633	0.3287	0.3967
UMass Amherst T2T	0.4818	0.4087	0.4276	0.7094	0.5961	0.6262
iria	0.4251	0.4902	0.4443	0.6174	0.7290	0.6536
MTIFL	0.5662	0.5014	0.5172	0.7964	0.7186	0.7373
MTI	0.5510	0.5415	0.5325	0.7703	0.7647	0.7514
AttentionMesh	0.5627	0.5235	0.5290	0.7902	0.7396	0.7472
DeepMeSH	0.5643	0.5364	0.5366	0.7899	0.7555	0.7560
MeSHProbeNet	**0.5901**	**0.5561**	**0.5596**	**0.8123**	**0.7714**	**0.7760**

3.4 Experimental results

We show the comparison result of the proposed MeSHProbeNet model with the default MTI, MTI First Line indexing (MTIFL) (Aronson *et al.*, 2004), DeepMeSH (Peng *et al.*, 2016), AttentionMeSH (Jin *et al.*, 2018), iria (Ribadas *et al.*, 2014), UMass Amherst T2T, MeSHmallow and Access Inn MAIstro on the last test set of the 2018 BioASQ challenge. There are 15 test sets in total (one test set per week during the challenge) and the complete results are available on the challenge webpage (http://participants-area.bioasq.org/results/6a/) (please note that we used the name xgx in the challenge). The main difference between MTI and MTIFL is that MTIFL has higher precision by limiting its recommendation to a smaller number of MeSH terms, while MTI balances precision and recall, and achieves better F-measure.

The comparison results based on the flat measures of each model are reported in Table 1. The challenge allows each model to make at most 5 attempts to try out different settings, such as different initializations and parameters, as a significance test. Our model consistently achieves the best performance. To conserve space, we only show the best performance score of each model here. Interested readers may refer to the complete result on the challenge website. The best scores are highlighted in boldface in Table 1. Compared with MTI, MTIFL has higher precision but lower recall, resulting in low F-measures. DeepMeSH outperforms MTI in terms of MiF score but its MaF score is not as good as MTI's, which means DeepMeSH pays more attention to the frequent MeSH terms such as 'humans', 'animals', 'male' and 'female'. We can observe that MeSHProbeNet achieves the highest scores in all F-Measures and accuracy. Since MeSHProbeNet is able to capture the correlation between different MeSH terms and MeSH indexing for infrequent terms can benefit from this correlation information, MeSHProbeNet gains both the best MiF and the best MaF scores.

The comparison results based on the hierarchical measures of each model are reported in Table 2. As with the flat measure result, we also only show the best performance score of each MeSH indexing model. The best scores are highlighted in boldface. The hierarchical measures are calculated based on the hierarchical structure of the MeSH vocabulary, thus the semantic distance between MeSH terms is under consideration. As with their performances on the flat measures, MTI achieves higher F-Measures than MTIFL and DeepMeSH outperforms both of them. We can see that MeSHProbeNet obtains the highest scores in all measures.

3.5 Ablation studies on MeSH probes

We have demonstrated strong empirical results of MeSHProbeNet. Now we perform ablation experiments in order to better understand the importance of the self-attentive MeSH probes. Since the 2018 BioASQ challenge is closed and the challenge test sets are currently not available, we split the dataset into training and test sets. The test set contains 7000 articles and is used to evaluate the ablation models. All the models are trained on this new training set.

To show the effect of MeSH probes, we include in the comparison bi-GRU, which directly feeds the GRU output to the multi-view neural classifier and uses no MeSH probes. To show the influence of different numbers of MeSH probes, MeSHProbeNet models with 5, 15 and 25 MeSH probes are also included in the comparison, among which the MeSHProbeNet-25 model has the same amount of MeSH probes as the model we used in the challenge. All the other parameters, such as the embedding dimension and the number of GRU layers, are the same as the challenge model for each model.

The ablation results based on the flat measures and the hierarchical measures are reported in Tables 3 and 4, respectively. The best scores are highlighted in boldface. One can observe that the self-attentive MeSH probe mechanism significantly improves the performance. Adding more MeSH probes is also helpful, although the improvement per added MeSH probe becomes less and less significant as the number of MeSH probes gets higher. Adding more probes will also increase the computation cost and disk usage of the model.

3.6 Computational efficiency

The training of MeSHProbeNet on the entire MEDLINE database can be finished within 24 hours with one NVIDIA TITAN Xp GPU. Given a new test set of 10 000 articles, the prediction takes less than 1 minute. Compared with other state-of-the-art MeSH indexing models, for example, DeepMeSH needs 1 week to train on 1 million articles and AttentionMeSH needs 4 days to train on 3 million articles with 2 GPUs, this improved training efficiency of MeSHProbeNet allows us to exploit the entire database of more than 13 million annotated articles. Moreover, since MeSHProbeNet does not need to store any article information to perform KNN to find similar articles in the database, nor does it need to train

Table 3. Ablation results based on the flat measures

Models	MiP	MiR	MiF	EBP	EBR	EBF	MaP	MaR	MaF	Acc
bi-GRU	0.6691	0.6243	0.6459	0.6701	0.6357	0.6331	0.6228	0.4997	0.4937	0.4801
MeSHProbeNet-5	0.6978	0.6511	0.6736	0.6975	0.6660	0.6628	0.6500	0.5609	0.5485	0.5124
MeSHProbeNet-15	0.7072	0.6617	0.6837	0.7073	0.6770	0.6732	0.6675	0.5792	0.5670	0.5243
MeSHProbeNet-25	**0.7094**	**0.6643**	**0.6861**	**0.7092**	**0.6801**	**0.6760**	**0.6732**	**0.5846**	**0.5706**	**0.5276**

Table 4. Ablation results based on the hierarchical measures

Models	LCA-P	LCA-R	LCA-F	HiP	HiR	HiF
bi-GRU	0.5610	0.5111	0.5200	0.7935	0.7250	0.7380
MeSHProbeNet-5	0.5767	0.5341	0.5400	0.8064	0.7487	0.7583
MeSHProbeNet-15	0.5844	0.5434	0.5487	0.8118	0.7575	0.7660
MeSHProbeNet-25	**0.5859**	**0.5465**	**0.5511**	**0.8129**	**0.7606**	**0.7681**

separate classifiers for more than 28 000 MeSH terms, the disk usage of MeSHProbeNet is just about 1 GB.

4 Conclusion

We present an end-to-end MeSH indexing model MeSHProbeNet. MeSHProbeNet participated in the 2018 BioASQ challenge and achieved the best performance in the latest batch. MeSHProbeNet is a self-attentive deep neural network classifier, which is able to extract different aspects of biomedical knowledge from an input article with different MeSH probes, and generate MeSH recommendations based on the extracted features, journal information and MeSH correlations. The experimental results demonstrate the effectiveness of MeSHProbeNet on both frequent and infrequent MeSH terms.

Funding

This work was supported by the US National Science Foundation under grant NSF IIS-1514204. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

Conflict of Interest: none declared.

References

Aronson,A.R. and Lang,F. (2010) An overview of metamap: historical perspective and recent advances. *JAMIA*, **17**, 229–236.

Aronson,A.R. *et al.* (2004) The NLM indexing initiative's medical text indexer. In: Fieschi,M. *et al.* (eds.) *MEDINFO 2004—Proceedings of the 11th World Congress on Medical Informatics, San Francisco, California, USA, September 7–11, 2004, Volume 107 of Studies in Health Technology and Informatics*. IOS Press, Amsterdam, The Netherlands, pp. 268–272.

Bahdanau,D. *et al.* (2014) Neural machine translation by jointly learning to align and translate. https://dblp.org/rec/bib/journals/corr/BahdanauCB14.

Bengio,Y. *et al.* (2006) Neural probabilistic language models. In: Holmes,D.E. and Jain,C.L. (eds) *Innovations in Machine Learning*. Springer, Berlin, Heidelberg, pp. 137–186.

Cho,K. *et al.* (2014) On the properties of neural machine translation: encoder-decoder approaches. In: Wu,D. *et al.* (eds) Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014, pp. 103–111.

Chung,J. *et al.* (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. https://dblp.org/rec/bib/journals/corr/ChungGCB14.

Coordinators,N.R. (2016) Database resources of the national center for biotechnology information. *Nucleic Acids Res.*, **44**, D7.

Gopalakrishnan,V. *et al.* (2018) Towards self-learning based hypotheses generation in biomedical text domain. *Bioinformatics*, **34**, 2103–2115.

Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.

Huang,G. *et al.* (2017) Snapshot ensembles: train 1, get M for free. https://dblp.org/rec/bib/journals/corr/HuangLPLHW17.

Jha,K. *et al.* (2017) Augmenting word embeddings through external knowledge-base for biomedical application. In: Jian-Yun,N, *et al*, (eds) *2017 IEEE International Conference on Big Data (Big Data)*. IEEE Computer Society, Boston, MA, USA, pp. 1965–1974.

Jin,Q. *et al.* (2018) Attentionmesh: Simple, effective and interpretable automatic mesh indexer. In: *Proceedings of the 6th BioASQ Workshop A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering*, Association for Computational Linguistics, Brussels, Belgium, pp. 47–56.

Kosmopoulos,A. *et al.* (2015) Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining Knowl. Disc.*, **29**, 820–865.

LeCun,Y. *et al.* (2015) Deep learning. *Nature*, **521**, 436.

Lin,J.J. and Wilbur,W.J. (2007) Pubmed related articles: a probabilistic topic-based model for content similarity. *BMC Bioinformatics*, **8**, 423.

Lin,Z. *et al.* (2017) A structured self-attentive sentence embedding. https://dblp.org/rec/bib/journals/corr/LinFSYXZB17.

Liu,K. *et al.* (2015) Meshlabeler: improving the accuracy of large-scale mesh indexing by integrating diverse evidence. *Bioinformatics*, **31**, 339–347.

Mikolov,T. *et al.* (2013) Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Vol. 2. Curran Associates Inc., Lake Tahoe, Nevada, USA, pp. 3111–3119.

Mork,J.G. *et al.* (2013) The NLM medical text indexer system for indexing biomedical literature. In: Ngomo,A.N. and Paliouras,G. (eds.) *Proceedings of the First Workshop on Bio-Medical Semantic Indexing and Question Answering, a Post-Conference Workshop of Conference and Labs of the Evaluation Forum 2013 (CLEF 2013), Valencia, Spain, September 27th, 2013., Volume 1094 of CEUR Workshop Proceedings*. CEUR-WS.org.

Mork,J.G. *et al.* (2014) Recent enhancements to the NLM medical text indexer. In: Cappellato,L. *et al.* (eds.) *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15–18, 2014, Volume 1180 of CEUR Workshop Proceedings*. CEUR-WS.org, pp. 1328–1336.

Nelson,S.J. *et al.* (2004) The MeSH translation maintenance system: structure, interface design, and implementation. In: Marius,F. (eds) *MEDINFO 2004 - Proceedings of the 11th World Congress on Medical Informatics, San Francisco, California, USA, September 7–11, 2004*, Vol. 107. IOS Press, pp. 67–69.

Peng,S. *et al.* (2016) Deepmesh: deep semantic representation for improving large-scale mesh indexing. *Bioinformatics*, **32**, 70–79.

Ribadas,F.J. *et al.* (2014) CoLe and UTAI participation at the 2014 BioASQ semantic indexing challenge. In: Linda,C. (eds) W*orking Notes for {CLEF} 2014 Conference, Sheffield, UK, September, 15–18, 2014*, CEUR-WS.org, pp. 1361–1374.

Tang,L. *et al.* (2009) Large scale multi-label classification via metalabeler. In: Juan,Q. (eds) *Proceedings of the 18th International Conference on World Wide Web WWW 2009, Madrid, Spain, April 20–24, 2009*. ACM, pp. 211–220. https://dblp.org/rec/bib/conf/www/2009.

Tsatsaronis,G. *et al*. (2015) An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, **16**, 138.

Vaswani,A. *et al.* (2017) Attention is all you need. In: Isabelle,G. (eds) *Advances in Neural Information Processing Systems* 2017, 4–9 December 2017, Long Beach, CA, USA, ACM, pp. 5998–6008. https://dblp.org/rec/bib/conf/nips/2017.

Xun,G. *et al*. (2017a) Collaboratively improving topic discovery and word embeddings by coordinating global and local contexts. In: Proceedings of the 23rd (ACM) (SIGKDD) International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13–17, 2017. ACM, pp. 535–543. doi:10.1145/3097983.

Xun,G. *et al*. (2017b) A correlated topic model using word embeddings. In: Carles,S. (ed.) *Proceedings of the 26th International Joint Conference on Artificial Intelligence, (IJCAI) 2017, Melbourne, Australia, August, 19–25, 2017*, ijcai.org, pp.4207–4213. https://dblp.org/rec/bib/conf/ijcai/2017.

Xun,G. *et al*. (2017c) Generating medical hypotheses based on evolutionary medical concepts. In: Vijay,R. (eds) *2017 IEEE International Conference on Data Mining (ICDM) 2017, New Orleans, LA, USA, November 18-21, 2017*, IEEE Computer Society, pp. 535–544. https://dblp.org/rec/bib/conf/icdm/2017.

Yuan,Y. *et al*. (2017) Wave2vec: learning deep representations for biosignals. In: Vijay,R. (eds) *2017 IEEE International Conference on Data Mining (ICDM) New Orleans, LA, USA, November 18–21, 2017*. IEEE Computer Society, pp. 1159–1164. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8211002.

Yuan,Y. *et al*. (2018) MuVAN: a multi-view attention network for multivariate temporal data. In: *2018 IEEE International Conference on Data Mining (ICDM)2018, Singapore, November 17–20, 2018*. IEEE Computer Society, pp. 717–726. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8591042.