AttentionMeSH: Simple, Effective and Interpretable Automatic MeSH Indexer

Qiao Jin*
University of Pittsburgh
qiao.jin@pitt.edu

Bhuwan Dhingra*
Carnegie Mellon University
bdhingra@cs.cmu.edu

William W. Cohen†
Google, Inc.
wcohen@google.com

Xinghua Lu
University of Pittsburgh
xinghua@pitt.edu

Abstract

There are millions of articles in PubMed database. To facilitate information retrieval, curators in the National Library of Medicine (NLM) assign a set of Medical Subject Headings (MeSH) to each article. MeSH is a hierarchically-organized vocabulary, containing about 28K different concepts, covering the fields from clinical medicine to information sciences. Several automatic MeSH indexing models have been developed to improve the time-consuming and financially expensive manual annotation, including the NLM official tool – Medical Text Indexer, and the winner of BioASQ Task5a challenge – DeepMeSH. However, these models are complex and not interpretable. We propose a novel end-to-end model, AttentionMeSH, which utilizes deep learning and attention mechanism to index MeSH terms to biomedical text. The attention mechanism enables the model to associate textual evidence with annotations, thus providing interpretability at the word level. The model also uses a novel masking mechanism to enhance accuracy and speed. In the final week of BioASQ Chanllenge Task6a, we ranked 2nd by average MiF using an on-construction model. After the contest, we achieve close to state-of-the-art MiF performance of ~ 0.684 using our final model. Human evaluations show AttentionMeSH also provides high level of interpretability, retrieving about 90% of all expert-labeled relevant words given an MeSH-article pair at 20 output.

1 Introduction

MEDLINE is a database containing more than 24 million biomedical journal citations by 2018[1]. PubMed provides free access to MEDLINE for worldwide researchers. To facilitate information storage and retrieval, curators at the National Library of Medicine (NLM) assign a set of Medical Subject Headings (MeSH) to each article. MeSH[2] is a hierarchically-organized terminology developed by NLM for indexing and cataloging biomedical texts like MEDLINE articles. MeSH has about 28 thousand terms by 2018[3], covering the fields from clinical medicine to information sciences. Indexers examine the full article and annotate it with MeSH terms according to rules set by NLM[4]. Its estimated that indexing an article costs $9.4 on average (Mork et al., 2013), and there are more than 813,500 citations added to MEDLINE in 2017[5]. Indexing all citations manually would cost several million dollars in one year. Thus, several automatic annotation models have been developed to improve the time-consuming and financially expensive manual annotation. We will discuss these models in section 2.1.

Automatical annotating PubMed abstracts with MeSH terms is hard in several aspects: There are 28 thousand possible classes and even more of their combinations. The frequencies of different MeSH terms also vary a lot: The most frequent MeSH term is 'Humans' and it is annotated to more than 8 million articles in the MEDLINE database; while the 20,000th frequent MeSH 'Hypnosis, Anesthetic' is indexed to only about 200 articles (Peng et al., 2016). It causes severe class imbalance problems. Above difficulties are further complicated by the fact that indexers at the NLM usually inspect the whole articles to

*These authors contribute equally to the paper.
†This work was done while the author was at CMU.
[1] https://www.nlm.nih.gov/pubs/factsheets/medline.html
[2] https://www.nlm.nih.gov/mesh
[3] https://www.nlm.nih.gov/pubs/factsheets/mesh.html
[4] https://www.nlm.nih.gov/bsd/indexing/training/TIP_010.html
[5] https://www.nlm.nih.gov/bsd/bsd_key.html

Proceedings of the 2018 EMNLP Workshop BioASQ: Large-scale Biomedical Semantic Indexing and Question Answering, pages 47–56
Brussels, Belgium, November 1st, 2018.

do the annotation, but the challenge only provides PubMed abstracts and titles, which might not be enough to find all MeSH terms. We will discuss it more detailedly in section 5.

Deep learning is a subtype of machine learning that arranges the computational models in multiple processing layers to learn the representations of data with multiple levels of abstractions as well as the mapping from these features to the output (LeCun et al., 2015). Attention is a strategy for deep learning models to learn both the mapping from input to output and the relevance between input parts and output parts (Bahdanau et al., 2014). The learnt relevance helps improve the mapping performance as well as provide interpretability. We will discuss relevant works of deep learning in automatic annotations in section 2.2.

Here we propose a novel model, AttentionMeSH, which utilizes deep learning and attention mechanism to index MeSH terms to biomedical texts and provides interpretation at the word level. Each abstract, together with title and journal name, is tokenized to words, then the model feeds word vectors to a bidirectional gated recurrent unit (BiGRU) to derive word representations with contextual information (Schuster and Paliwal, 1997; Cho et al., 2014). We narrow down the MeSH term vocabulary for each abstract using a masking mechanism. Then for each candidate MeSH term, the model calculates the attention attribution over words. Next, each MeSH term gets a specific document representations by MeSH-specific attention-weighted sum of the word vectors. Finally, the model uses nonlinear layers to classify each MeSH term using the learnt MeSH-specific document representation.

We participated in BioASQ Challenge Task6A while developing the model. We achieve close to state-of-the-art performance with an on-construction model in the final week of the contest and with our final model after the contest. The model also achieves high level of interpretability evaluated by human experts.

The main contributions of this work are summarized as follows:

1. To the best of our knowledge, AttentionMeSH is the first end-to-end deep learning model with soft-attention mechanism to index MeSH terms in such a large scale (millions of training data). With this relatively simple model, we achieved close to state-of-the-art performance without any sophisticated feature engineering or preprocessing.

2. We develop a novel masking mechanism, which is aimed to handle multi-class classification problems with a large number of classes, like indexing MeSH. We also conduct extensive experiments on how the masking layer settings influence classification performance.

3. We believe AttentionMeSH is the first MeSH annotation model that is capable of providing textual evidence and interpretations of its predictions. We argue that interpretability matters because humans are needed to complete the annotation task.

2 Related Work

2.1 Automatic MeSH Indexing

NLM developed Medical Text Indexer (MTI), a software for providing human indexers with automatic MeSH recommendations (Aronson et al., 2004). MTI takes as input a title and corresponding abstract to generate a set of recommended MeSH terms. MTI has two steps: the first is to generate MeSH candidates for recommendation, and the second is to filter and rank the candidates to give a final output. MTI uses MetaMap and nearest neighbor methods. MetaMap is another NLM-developed tool, which maps mentions in biomedical texts to Unified Medical Language System concepts (Aronson, 2001).

BioASQ is an European Union-funded project that organizes tasks on biomedical semantic indexing and question answering (Tsatsaronis et al., 2015). In the task A of BioASQ, participants are asked to annotate un-indexed PubMed articles with MeSH terms using their models, before they are annotated by the human indexers. The manual annotations are taken as ground truth to evaluate the participating models. DeepMeSH (Peng et al., 2016) is the winner of the latest challenge, BioASQ task 5a, held in 2017. DeepMeSH also uses a two-step strategy: the first step is to generate MeSH candidates and predict the number of output MeSH terms, and the second step is to rank the candidates and take the highest-ranked predicted number of MeSH terms as output. DeepMeSH uses Term Frequency Inverse Document Frequency (TFIDF) and document to vector (D2V) schemes to represent each abstract

and generate MeSH candidates using binary classifiers and k-nearest neighbor (KNN) methods over using these features. TFIDF is a traditional weighted bag of word sparse representation of the text and D2V learns a deep semantic representation of the text.

Because state-of-the-art models have less than 0.7 Micro-F, automatic MeSH indexing systems can just serve to assist human indexers. Since human indexers usually add or delete MeSH terms based on the recommendations, *interpretability* of the automatic annotations is very important for them. In this paper we adopt a *local explanation* view of model interpretability (Lipton, 2016), and argue that a good system, in addition to being accurate, should also be able to tell which part of the input supports the indexed MeSH term. This would allow human indexers to be more effective at annotating the article.

2.2 Deep Learning for Text Classification

Automatic indexing of MeSH terms to PubMed articles is a multi-label text classification problem. FastText (Joulin et al., 2016) is a simple and effective method for classifying texts based on n-gram embeddings. (Kim, 2014) used Convolutional Neural Networks (CNNs) for sentence-level classification tasks with state-of-the-art performance on 4 out of 7 tasks they tried. Very deep architectures such as that of (Conneau et al., 2017) have also been proposed for text classification. Motivated by these works we use an RNN-based model for classifying each MeSH term as being a positive label for a given article. We further use attention mechanism to boost performance and provide word-level interpretability.

Recently, there has been work on automatic annotation of International Classification of Diseases codes from clinical texts. (Shi et al., 2017) used character-level and word-level Long Short-Term Memory netowrks to get the document representations and (Mullenbach et al., 2018) used word-level 1-D CNN to get the document representations. Both these works utilized a soft attention strategy where each class gets a specific document represetation by weighted sum of the attention over words or phrases. Mullenbach et al. (2018) also highlighted the need for interpretability when annotating medical texts – in this work we apply similar ideas to the domain of MeSH indexing.

3 Methods

The model architecture is visualized in Figure 1. Starting from an input abstract, title and journal name, words in the document are embedded and fed to BiGRU to derive context-aware representations; KNN-derived articles from training corpus are identified and frequent MeSH terms in them are included as candidate annotations for the document. MeSH terms are embedded, and only those candidates are further considered in attention mechanism. We call it a masking mechanism. We apply an attention mechanism to assign attention weights to each word with respect to each candidate MeSH term, which leads to a MeSH-specific document representation. Finally, we use MeSH-specific document representations as input to perform classifications. For each candidate MeSH term of a document, the model outputs a probability. Binary cross-entropy loss is used for a gradient-based method to optimize the parameters. At inference time, the sigmoid outputs are converted to binary variables by thresholding.

3.1 Document Representation

For each article to be indexed, we first tokenize the journal name, title and abstract to words. In order to use the pre-trained word embeddings[6] provided by BioASQ organizer, we use the same tokenizer as they did. The pre-trained word embeddings are denoted as $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d_{e1}}$, where $|\mathcal{V}|$ is the vocabulary size and d_{e1} is the embedding size.

We can represent each article by a sequence of word embeddings corresponding to the tokenized text. The word embeddings are initialized by the BioASQ pre-trained word embeddings.

$$\mathbf{D} = \begin{bmatrix} \mathbf{w_1} & \dots & \mathbf{w_L} \end{bmatrix}^T \in \mathbb{R}^{L \times d_{e1}},$$

where L is the number of words in the journal name, title and abstract, and $\mathbf{w_i}$ is a vector for word at position i.

For each document representation $\mathbf{D}$, we feed this sequence of word vectors to an BiGRU to derive a context-aware sequence of word vectors:

$$\widetilde{\mathbf{D}} = \text{BiGRU}\left(\mathbf{D}\right) = \begin{bmatrix} \widetilde{\mathbf{w_1}} & \dots & \widetilde{\mathbf{w_L}} \end{bmatrix}^T \in \mathbb{R}^{L \times 2d_h},$$

where $\widetilde{\mathbf{w_i}}$ is the corresponding concatenated forward and backward hidden states of each word, and d_h is the hidden size of BiGRU.

[6] http://participants-area.bioasq.org/tools/BioASQword2vec/

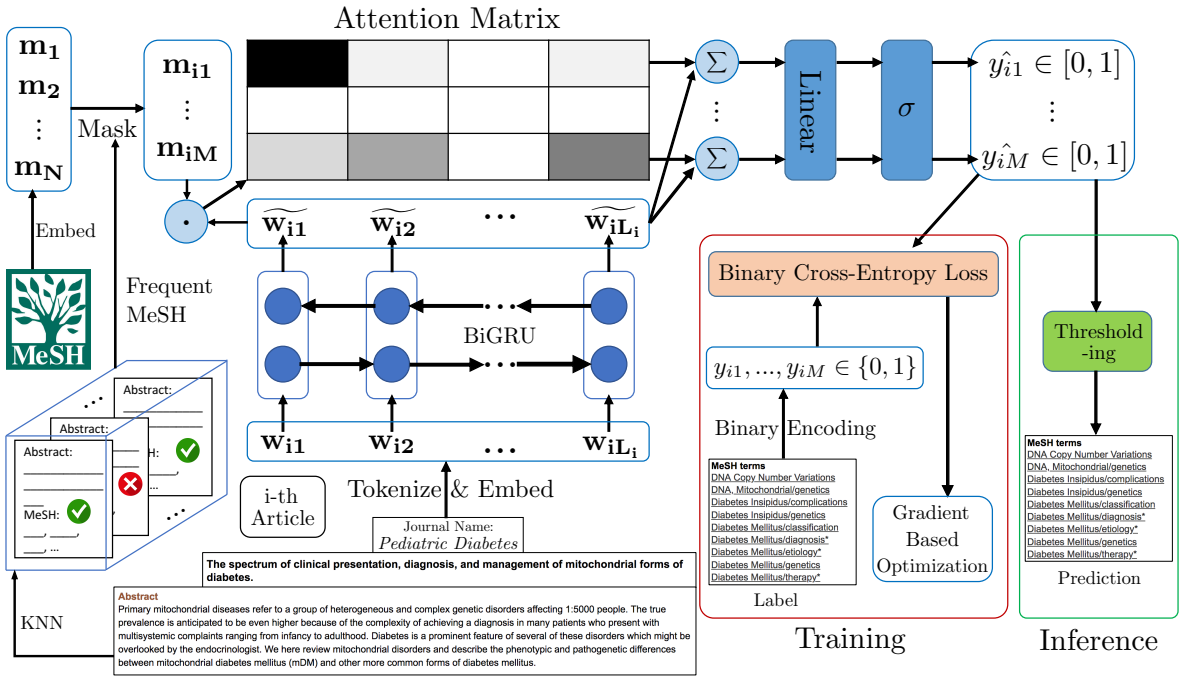

Figure 1: Model Architecture. BiGRU: Bi-directional recurrent gated unit. The example abstract is from (Karaa and Goldstein, 2015).

3.2 MeSH Representation and Masking

We learn the MeSH embedding matrix $\mathbf{H} \in \mathbb{R}^{N \times d_{e2}}$, where N is the number of all MeSH terms (28,340), and d_{e2} is the embedding size. For each article, we consider only a subset of all 28k MeSH terms for two reasons: 1. For each MeSH term, there are far more negative samples than the positive ones. We achieve down-sampling of the negative samples by considering only a subset of all MeSH terms as candidate for each article, so that the classifier only concentrate on choosing a most suitable MeSH among a set of plausible annotations; 2. It's more time efficient than training all the MeSH terms or training the MeSH classifiers one by one. We call it a masking layer.

We use KNN strategy to choose a specific subset of MeSH terms to train for each article:

Each abstract can be represented by IDF-weighted sum of word vectors:

$$\mathbf{d} = \frac{\sum_{i=1}^{n} IDF_i \times \mathbf{w_i}}{\sum_{i=1}^{n} IDF_i} \in \mathbb{R}^{d_e 1},$$

where $\mathbf{w_i}$ is the corresponding word vector, and IDF_i is the inverse document frequency of this word.

We then calculate cosine similarity of representations between the abstracts:

$$\text{Similarity}(i,j) = \frac{\mathbf{d_i^T d_j}}{||\mathbf{d_i}|| \times ||\mathbf{d_j}||}$$

For each article, we find its K nearest neighbors based on cosine similarity. And then we count the MeSH term frequency in these neighbors. The most frequent M MeSH terms are trained for each article. We denote the masked MeSH embedding as $\mathbf{H'}$,

$$\mathbf{H'} = \begin{bmatrix} \mathbf{m_1} & \mathbf{m_2} & \dots & \mathbf{m_M} \end{bmatrix} \in \mathbb{R}^{M \times d_{e2}},$$

where we make $d_{e2} = d_h$ so that we could directly get the dot product of each MeSH representation and word vector.

3.3 Attention Mechanism

After getting the document representation and masked MeSH representations, we calculate the dot products between each context-aware word vector and each MeSH embedding, which represents the similarity within each pair:

$$\mathbf{S} = \mathbf{H'}\,\widetilde{\mathbf{D}}^T = \begin{bmatrix} \widetilde{\mathbf{D}}\,\mathbf{m_1} & \dots & \widetilde{\mathbf{D}}\,\mathbf{m_M} \end{bmatrix}^T \in \mathbb{R}^{M \times L},$$

We then uses SoftMax function to normalize over the word axis to get attention weights attribution for each MeSH term:

$$\text{SoftMax}\,(\mathbf{Sim})$$

$$= \left[\text{SoftMax}(\widetilde{\mathbf{D}}\,\mathbf{m_1}) \quad ... \quad \text{SoftMax}(\widetilde{\mathbf{D}}\,\mathbf{m_M})\right]^T$$

$$= \left[\boldsymbol{\alpha_1} \quad ... \quad \boldsymbol{\alpha_M}\right]^T \in [0,1]^{M\times L},$$

where $\boldsymbol{\alpha_j} \in [0,1]^L$ is the attention weights over words for MeSH term j, and $\sum_{k=1}^{L} \alpha_{jk} = 1$.

3.4 Classification

For each MeSH term, we can have a MeSH-specific representation of document by sum of word vectors weighted by attention weights:

$$\mathbf{R_j} = \boldsymbol{\alpha_j}\widetilde{\mathbf{D}} \in \mathbb{R}^{2d_h},$$

where $\mathbf{R_j}$ is MeSH term j specific document representation. We apply a linear projection layer and sigmoid activatin function to each MeSH term, finally getting the output probability:

$$\hat{y_j} = \sigma(\mathbf{R_j^T m_j'} + b_j) \in [0,1],$$

where $\mathbf{m_j'}$ and b_j are learnable linear projection parameters for MeSH term j. We model

$$P(\text{MeSH j indexed} \mid \text{Journal, Title, Abstract}) = \hat{y_j}.$$

3.5 Training

After get the conditioned probability we model, we can calculate the binary cross-entropy loss for each MeSH term:

$$\mathcal{L}_j = -(y_j log(\hat{y_j}) + (1-y_j) log(1-\hat{y_j})),$$

where $y_j \in \{0,1\}$ is the ground-truth label of MeSH j. $y_j = 0$ means MeSH j is not annotated to the article by human indexers, while $y_j = 1$ means MeSH j is annotated. We can get the total loss by summing them up:

$$\mathcal{L} = \frac{1}{M}\sum_{j=1}^{M}\mathcal{L}_j$$

The model is trained end-to-end from word and MeSH embedding to the final projection layer by a gradient-based optimization algorithm to minimize $\mathcal{L}$.

3.6 Inference

At inference time, we will predict the MeSH terms whose predicted probability is larger than a tuned threshold:

$$(\text{predict MeSH j}) = \mathbb{1}(\hat{y_j} > p_j),$$

where p_j is the tuned threshold for MeSH term j. The thresholds are tuned to maximize MiF:

$$p_1, ..., p_N = \underset{p_1,...,p_N}{\text{argmax}}\, \text{MiF}(\text{Model}, p_1, ..., p_N)$$

We tune p by the the micro-F optimization algorithm described in (Pillai et al., 2013), which they proved to be able to achieve the global maximum.

4 Experiments

4.1 Dataset

We use the dataset provided by BioASQ[7], which contains about 13.5 million manually annotated PubMed articles. The dataset covers 28,340 MeSH terms in total, and each article is annotated 12.69 MeSH terms on average. We selected 3 million articles from 2012 to 2017 for training.

The results reported in this paper are derived from two test sets: **BioASQ Test Sets**: During the challenge, BioASQ provides a test set of several thousands articles each week. **Ours**: we use 100 thousand latest articles to test our model, and all other results are calculated by this dataset. Since our test set is very large, the results will be precise.

4.2 Configuration

The model is implemented using PyTorch (Paszke et al., 2017). The parameter settings are shown in Table 1. We use Adam optimizer and batch size of 32. We train 2 epochs of each model on the 3M article training set, and apply hyperbolic learning rate decay and early stopping strategies (Yao et al., 2007). The training takes 4 days on 2 GPUs (GeForce GTX TITAN X). Before tuning the thresholds for all individual MeSH term, we use a global threshold of 0.35 due to the highly imbalanced dataset.

4.3 Evaluation Metric

The major metric for performance evaluation is Micro-F, which is a harmonic mean of micro-precision (MiP) and micro-recall (MiR) , and is calculated as follows:

$$\text{Micro-F} = \frac{2 \cdot \text{MiP} \cdot \text{MiR}}{\text{MiP} + \text{MiR}},$$

where

$$\text{MiP} = \frac{\sum_{i=1}^{N_a}\sum_{j=1}^{N} y_{ij} \cdot \hat{y_{ij}}}{\sum_{i=1}^{N_a}\sum_{j=1}^{N} \hat{y_{ij}}}$$

[7] http://participants-area.bioasq.org/general_information/Task6a/

Parameter	Value(s)
$\|\mathcal{V}\|$	1.7M
d_{e1}	256
d_{e2}	512
d_h	256
N	28,340
L	$\leq$512 (truncated if longer)
N_a BioASQ	5,833$\sim$10,488
N_a Ours	100,000
K	0.1k, 0.5k, **1k**, 3M
M	128, 256, 512, **1,024**
Learning Rate	0.002, **0.001**, 0.0005
BiGRU Layer(s)	1, 2, **3**, 4

Table 1: Parameter Values. For hyperparameters, we highlight the optimal ones among all tried values.

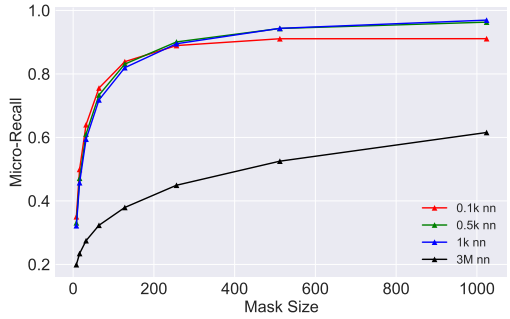

Figure 2: The micro-recall of MeSH terms versus different mask sizes for different numbers of neighbor articles.

$$\text{MiR} = \frac{\sum_{i=1}^{N_a}\sum_{j=1}^{N} y_{ij}\cdot\hat{y_{ij}}}{\sum_{i=1}^{N_a}\sum_{j=1}^{N} y_{ij}}$$

In these equations, i is indexed for articles and j is indexed for MeSH terms, so N_a is the number of articles in the test set, and N is the number of all MeSH terms. y_{ij} and $\hat{y_{ij}}$ are both binary encoded variables to denote whether MeSH term j is in article i in ground-truth and prediction, respectively.

4.4 Evaluation of Masking Layer

Selecting relevant MeSH terms from neighbor articles can be regarded as a weak classifier itself, and high-recall setting is favored in this step. We measure the micro-recall for different masking layer settings, and the results are shown in Figure 2. Basically, there are two hyperparameters for it: the number of neighbor articles K and the number of highest ranking MeSH terms selected M. A non-trivial baseline for K is 3M, i.e. the number of all training articles. Under this circumstance, the ranked MeSH list is determined by global frequency, thus is non-specific to any article.

We choose the number of nearest articles $K = 1000$ for it gives the highest recalls with the increase of mask size. In fact, micro-recall at $M = 1024$ and $K = 1000$ is about 0.97, which almost guarantees that all true annotations are included as candidate for a document. Before fine-tuning on other hyperparameters and the thresholds of making predictions, we first train the model with different M, and report the results in Table 2.

Mask Setting	MiP	MiR	MiF
1,024 rd.	0.5891	0.0173	0.0337
1,024 freq.	**0.6863**	0.4257	0.5262
128 n.n.	0.6354	0.5880	0.6108
256 n.n.	0.6690	0.5975	0.6312
512 n.n.	0.6663	0.6116	0.6378
1,024 n.n.	0.6698	**0.6262**	**0.6472**

Table 2: Model Performance with Different Mask Settings. n.n.: MeSH mask selected from nearest neighbor articles ($K = 1000$); freq.: MeSH mask selected from globally frequent MeSH terms; rd.: MeSH mask randomly selected. All results are averaged over models trained by 3 random seeds.

4.5 Evaluation of Performance

While we were developing the model, we participated in the BioASQ Task6a challenge. During the challenge, there is a test set available each week. Each test set contains several thousands of un-indexed PubMed citations. Each citation has journal name, title, abstract information. Participants will run their models on the test set and upload their predictions of MeSH annotations within a given time. The organizers will then evaluate every participants' predictions and make the results available. The results of the whole Challenges are showed in Figure 3. Furthermore, the results of the last week of the Challenge are showed in Table 3.

Model	Average MiF	Maximum MiF
Access Inn MAIstro	0.2788	0.2788
MeSHmallow	0.3161	0.3161
UMass Amherst T2T	0.4988	0.4988
iria	0.4992	0.5161
MTI First Line Index	0.6332	0.6332
DeepMeSH	0.6451	0.6637
Default MTI	0.6474	0.6474
AttentionMeSH	0.6635	0.6635
xgx	**0.6862**	**0.6880**

Table 3: Model Performance of the Final BioASQ Test Set. The models are ranked top-down from the lowest average MiF to the highest one. Our on-construction AttentionMeSH ranked second by average MiF.

It should be noted that the models we used in

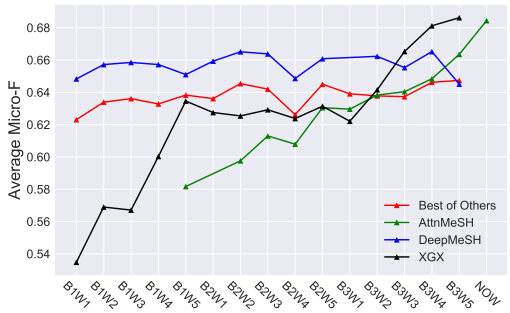

Figure 3: BioASQ Challenge Task6A results. BaWb: Test week b of batch a. From B1W1 to B3W5, we show the average MiF of different models: AttentionMeSH, DeepMeSH, XGX, and the best performance of all other models. Results are retrieved from http://participants-area.bioasq.org/results/6a/ on June 10th, 2018. And NOW shows the most up-to-date results from our test set.

BioASQ are not our final model. We include the up-to-date results in 'NOW' in Figure 3 and report the ablation test results in Table 4.

Model	MiP	MiR	MiF
AttentionMeSH (AM)	0.6698	0.6262	0.6472
AM w/o BiGRU	0.6362	0.5848	0.6093
AM w/o learning w.e.	0.6657	0.6106	0.6369
AM w/o attention	0.6807	0.5519	0.6095
AM w/ t.t.	0.7048	0.6393	0.6704
AM w/ ensemble & t.t.	**0.7172**	**0.6543**	**0.6844**

Table 4: Model Performance with Ablations and Finer Tuning. w/o: without; w/: with; t.t.: MeSH term threshold tuning; w.e.: word embeddings. Ensembling takes the average prediction of 8 models trained by different seeds. All results, except the ensemble one, are averaged over models trained by 3 random seeds.

4.6 Evaluation of Interpretability

At inference time, attention matrix provides word-level interpretation: For each MeSH prediction, the model shows which words are given high attention. It helps the indexers to evaluate and proofread the indexing results of our model. Figure 4 shows an example of attention for interpretation.

To qualitatively evaluate the interpretability of different models, the best way would be to measure the time efficiency of manual indexing with the assistance of different models. However, this might require well-trained NLM indexers to evaluate. Instead, we asked two independent researchers with Ph.D. degrees in related fields to label relevant words for 100 MeSH-article pairs. Their intersected labels are regarded as ground-truth. We model the interpretability evaluation as an information retrieval task, and evaluate each method's recall at different numbers of outputs in Table 5. Since other models like DeepMeSH and MTI don't report how to interpret their model outputs, we use string-matching as a non-trivial baseline.

Model	R@5	R@10	R@20
String-Matching	0.3890	0.4180	0.4336
AM Embeddings	0.6180†	0.7486†	0.8088†
AM Whole Model	**0.6929**†‡	**0.8389**†‡	**0.8993**†‡

Table 5: Interpretability Evaluation. R@n: The average recall of ground-truth relevant words if the model outputs n words. AM Whole Model: The whole model of AttentionMeSH is used to get the attention matrix, and n words with highest attention weights will be the output; AM Embeddings: We only use the trained word and MeSH embeddings of AttentionMeSH model, and we output n words that have highest dot products with each specific MeSH. String-Matching: A string matching method that takes all words in the abstracts that are same to any word in the MeSH name. †: Significant differences with String-Matching; ‡: Significant differences with AM Embeddings. Significance is defined by $p < 0.05$ in paired t tests.

5 Discussion

One intrinsic limitation of all present automatic MeSH indexing models, including us, is that these models just annotate MeSH terms from the abstract, title, journal name etc, but they don't look into the article bodies. However, the human indexers in NLM do need to look into the bodies to annotate each article, and thus the textual evidence for certain annotations is missed during training. As such, all present models won't have enough information to do the annotation, and certain percent of false negatives is inevitable, and the performance is upbounded by them. For example, MeSH terms 'Humans', 'Males', 'Females' are annotated to our demo article in Figure 4. However, the abstract doesn't contain any relevant information. 35 articles in our 100 MeSH-article pairs evaluated by experts don't have any words relevant to the MeSH term.

We noted that AttentionMeSH predicted many MeSH terms to documents that were not annotated by NLM indexers, which appears to be "false pos-

PLoS One

Association of SNP rs80659072 in the ZRS with polydactyly in Beijing You chickens

Abstract

The Beijing You chicken is a Chinese native breed with superior meat quality and a unique appearance. The G/T mutation of SNP rs80659072 in the Shh long-range regulator of GGA2 is highly associated with the polydactyly phenotype in some chicken breeds. In the present study, this SNP was genotyped using the TaqMan detection method, and its association with the number of toes was analyzed in a flock of 158 birds of the Beijing You population maintained at the Beijing Academy of Agriculture and Forestry Sciences. Furthermore, the skeletal structure of the digits was dissected and assembled in 113 birds. The findings revealed that the toes of Beijing You chickens were rich and more complex than expected. The plausible mutation rs80659072 in the zone of polarizing activity regulatory sequence (ZRS) in chickens was an essential but not sufficient condition for polydactyly and polyphalangy in Beijing You chickens. Several individuals shared the T allele but showed normal four-digit conformations. However, breeding trials demonstrated that the T allele could serve as a strong genetic marker for five-toe selection in Beijing You chickens.

True Positive MeSH: ***Toes***
- …with the number of **toes** was analyzed in a…
- …findings revealed that the **toes** of Beijing You chickens…
- …skeletal structure of the **digits** was dissected and assembled…

True Positive MeSH: ***Polymorphism, Single Nucleotide***
- Association of **SNP** rs80659072 in the ZRS…
- …The G/T mutation of **SNP** rs80659072 in the Shh…
- …this **SNP** was genotyped using the TaqMan…

False Positive MeSH: ***China***
- …ZRS with polydactyly in **Beijing** You chickens…
- …**Beijing** You chicken is a **Chinese** native breed with…

False Negative MeSH: ***Meat***
- …native breed with superior **meat** quality and…
- …The Beijing You **chicken** is a Chinese native breed…
- …in Beijing You **chickens**

Figure 4: Attention Display. In a randomly-selected test article (Chu et al., 2017), we show the 3 words that are given highest attention weights for 4 MeSH terms, including two true positive, one false positive and one false negative predictions.

itives". However, after manual inspection, we noticed that many of our predictions are semantically sensible. For example, both the articles in Figure 4 and Figure 5 discuss genotype-phenotype relationship in Beijing You chickens. However, MeSH term China is annotated to the article in Figure 5, but not the one in Figure 4. We conjecture that this may be due to inconsistency among indexers and that automatic indexing may assign more semantically sensible annotations to enhance the coverage of concepts in a document.

In consideration of the limitations and problems mentioned above, some false positive and false negative MeSH terms are unavoidable. We argue that human experts' performance on test dataset based on the same input as given in BioASQ is needed to provide better evaluation and comparison of performance of current methods.

Concerning how the explanations will help, we just perform a preliminary study by human evaluators, where we model the interpretability as an information retrieval (IR) task. However, the potential users don't regard the annotation task as an IR task. Thus, it would be more convincing to recruit some indexers at NLM and conduct a user study, measuring the annotation efficiency and accuracy with and without the help of AttentionMeSH.

Animal Biotechnology

The effect of a mutation in the 3-UTR region of the HMGCR gene on cholesterol in Beijing-you chickens.

Abstract

The 3-hydroxyl-3-methylglutaryl Coenzyme A reductase (HMGCR) gene was examined for polymorphisms in Beijing-you chickens. A "T" base insert was detected at nucleotide 2749 of the 3-UTR region of the HMGCR gene and was used as the basis for distinguishing a B allele, distinct from the A. Serum and muscle contents of total cholesterol. LDL-cholesterol in serum was significantly lower in AB birds and lowest in BB birds. Real-time PCR showed that the same trends across genotypes occurred in an abundance of HMGCR transcripts in liver, but there was no difference in contents of HMGCR mRNA in breast or thigh muscles. Hepatic expression and serum LDL-cholesterol were meaningfully correlated (partial, with total serum cholesterol held constant, r = 0.923). In muscle, similar genotypic differences were found for the abundance of the LDL receptor (LDLR) transcript. Cholesterol content in breast muscle related to LDLR expression (partial correlation with serum LDL-cholesterol held constant, r = 0.719); the equivalent partial correlation in thigh muscle was not significant. The results indicated that the B allele significantly reduces hepatic abundance of HMGCR transcripts, probably accounting for genotypic differences in serum cholesterol. In muscle, the cholesterol content appeared to reflect differences in LDLR expression with apparent mechanistic differences between breast and thigh.

Figure 5: A Contradictorily Indexed Article (Cui et al., 2010). MeSH term China is annotated to this article, while not to a similar one at Figure 4.

6 Conclusions

We present AttentionMeSH, an automatic MeSH indexer, which is simple and interpretable. It also achieves comparable performance to the current state-of-the-art. Since even the state-of-the-art model has only about 0.69 by MiF metric, manual annotations are still required. Thus, interpretability of the models is vital. We evaluate the interpretability of AttentionMeSH by retrieving capability of experts-labeled relevant words. Our model achieves high performance by this task. To the best of our knowledge, AttentionMeSH is the only interpretable model for indexing MeSH which has close to state-of-the-art performance.

7 Acknowledgement

This work has been supported by NIH grant No. 5R01LM012011. We thank Dr. Chunhui Cai and Dr. Lujia Chen for their independent evaluations of model interpretability. We are also grateful for the anonymous reviewers who gave us very insightful suggestions. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Alan R Aronson. 2001. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Alan R Aronson, James G Mork, Clifford W Gay, Susanne M Humphrey, Willie J Rogers, et al. 2004. The nlm indexing initiative's medical text indexer. *Medinfo*, 89.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Qin Chu, Zhixun Yan, Jian Zhang, Tahir Usman, Yao Zhang, Hui Liu, Haihong Wang, Ailian Geng, and Huagui Liu. 2017. Association of snp rs80659072 in the zrs with polydactyly in beijing you chickens. *PloS one*, 12(10):e0185953.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116.

HX Cui, SY Yang, HY Wang, JP Zhao, RR Jiang, GP Zhao, JL Chen, MQ Zheng, XH Li, and J Wen. 2010. The effect of a mutation in the 3-utr region of the hmgcr gene on cholesterol in beijing-you chickens. *Animal biotechnology*, 21(4):241–251.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Amel Karaa and Amy Goldstein. 2015. The spectrum of clinical presentation, diagnosis, and management of mitochondrial forms of diabetes. *Pediatric diabetes*, 16(1):1–9.

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

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436.

Zachary C Lipton. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

James G Mork, Antonio Jimeno-Yepes, and Alan R Aronson. 2013. The nlm medical text indexer system for indexing biomedical literature. In *BioASQ@ CLEF*.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*.

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

Shengwen Peng, Ronghui You, Hongning Wang, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2016. Deepmesh: deep semantic representation for improving large-scale mesh indexing. *Bioinformatics*, 32(12):i70–i79.

Ignazio Pillai, Giorgio Fumera, and Fabio Roli. 2013. Threshold optimisation for multi-label classifiers. *Pattern Recognition*, 46(7):2055–2065.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. 2017. Towards automated icd coding using deep learning. *arXiv preprint arXiv:1711.04075*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.