Knowledge-Based Systems 211 (2021) 106487

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

InPHYNet: Leveraging attention-based multitask recurrent networks for multi-label physics text classification

Vishaal Udandarao [a,1], Abhishek Agarwal [b,1], Anubha Gupta [c,*], Tanmoy Chakraborty [a]

[a] *Department of CSE, IIIT-Delhi, New Delhi, 110020, India*
[b] *Department of CSE, Department of Mathematics, IIIT-Delhi, New Delhi, 110020, India*
[c] *SBILab, Department of ECE, IIIT-Delhi, New Delhi, 110020, India*

ARTICLE INFO

Article history:
Received 15 April 2020
Received in revised form 17 July 2020
Accepted 25 September 2020
Available online 13 October 2020

Keywords:
Architectures for educational technology system
AI in physics education
Attention-based multitask recurrent network
Multi-label classification

ABSTRACT

The ability to create and sustain educational infrastructure is a major challenge to nations across the world. Today, information technology is increasingly being used to alleviate this problem by bridging the gap between learners and the textual materials by automating the process of teaching and learning. Due to this, there has been a steep rise in the information need for pedagogical content in recent years. Although there is increasing interest in building question-answering systems, there is a scarcity of intelligent tutoring systems, particularly, in physics education that can aid both students and teachers in secondary education. In this paper, we introduce a novel method for multi-label classification of paragraphs, where the paragraphs are chosen from physics subject of 6^{th} to 12^{th} grades from the curriculum of Central Board of Secondary Education (CBSE), India. This curriculum is common across India. For this purpose, we have constructed an attention-based recurrent interleaved multi-task learning (MTL) network, namely InPHYNet that can be used for any general purpose multi-label classification task related to the educational domain. The proposed solution is contextual and scalable. Although related to physics education, it is generalizable as an approach for other subjects. We perform experiments (i) to verify and validate the labels of data collected, and (ii) to conduct robust analysis of the proposed InPHYNet network. It is observed to yield significant accuracy on the dataset and can be used for any education-based text classification/annotation or as a module within the educational question-answering systems to enhance its quality.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Science, Technology, Engineering, and Mathematics (STEM) jobs are important in the growth of any nation. STEM workers drive this growth by generating new technological ideas, products and companies. According to the report by the US Department of Commerce [1], STEM areas experienced higher growth rates of 0.8% per year compared to a growth rate of 0.3% per year in non-STEM areas during the period 2005–2015. The report emphasized that this higher growth trend in STEM areas would continue in the next decade (2014–2024) with the projections of yearly employment growth of 0.9% in STEM areas compared to an yearly growth of 0.6% in non-STEM areas. Similarly, in 2015, STEM professionals were observed to earn 29% more compared to those working in non-STEM areas. It is widely known by now that preparedness of a country's population in STEM areas will influence its success in knowledge-based global economy because most of the future jobs will require basic mathematics and science skills. However, many students leave science education early in their careers [2,3]. This is partly due to difficulties in learning STEM subjects and partly due to challenges in teaching STEM related subjects [4]. There is definitely a need for new pedagogical practices for teaching STEM subjects. Many education researchers have proposed solutions to address these problems. Kelley and Knowles [5] presented a conceptual framework towards integrating STEM areas' key concepts with real life applications for generating interest in students. Moore et al. [6] discussed implementation and integration of engineering concepts in K-12 STEM education, whereas Wladis et al. [7] discussed online learning and factors affecting course retention in STEM.

Often, physics is viewed as one of the toughest subjects by students. It has been observed that the traditional teaching methods are not able to change this belief irrespective of the instructor or the quality of instruction [8,9]. Thus, students resort to rote memorization of concepts that eventually leads to lack of interest in physics. Hake [10] noted that students receiving instruction through traditional teaching mode can master, on an average,

[*] Corresponding author.
E-mail addresses: anubha@iiitd.ac.in (A. Gupta), tanmoy@iiitd.ac.in (T. Chakraborty).
[1] Equal contribution.

https://doi.org/10.1016/j.knosys.2020.106487
0950-7051/© 2020 Elsevier B.V. All rights reserved.

only 30% or less new concepts taught in the class. This result is also stated to be largely independent of lecturer quality, class size, or institution [10]. Cognitive research has shown that material presented in a classroom lecture is more than a typical person can process, leading to cognitive load and decreased information processing [11]. Wieman and Perkins [12] discussed how this cognitive load can be minimized by having an organized structure of presented ideas and by linking new material to the ideas already known to the audience. They also mentioned the importance of introducing concepts of physics in terms of real-world situations and how educational technology can greatly improve physics education by facilitating the incorporation of these principles into instruction.

In this paper, we present a study of using educational technology in a context where it augments the work of teachers rather than replacing teachers. Our core concept is to help teachers as they assist students in class with problem-solving and learning. We foresee a question-answering system, that is although automated, should partly be supported by experts available online on the platform, if needed. We are designing such a system, specifically, for physics subject of 6th to 12th grades on the syllabus as prescribed by the Central Board of Secondary Education (CBSE), India.

In this study, we undertake the first step towards the development of this system. Since there was no curated/annotated dataset available for CBSE physics content in the ready-to-use form that can be utilized to build a question-answering physics chatbot, we prepared an in-house annotated dataset for this purpose. A detailed description on the collection and annotation of the dataset, and the visualization and analysis of the dataset is provided in Section 3. In the near future, we aim to build an automated question-answering system for the educational (physics) domain, which would understand the correlation between various concepts and entities. For this, we first need to build a model that can classify text into the type of information it contains. For instance, the paragraph "*Force can change speed of a moving body: acceleration will increase speed of a running vehicle while applying brakes will decrease the speed of a running vehicle.*" belongs to two classes – "Effects" and "Examples". Along with building the physics dataset, we also propose an attention-based multitask recurrent network, called InPHYNet[2] that can be used for text classification. We also performed rigorous experiments on the dataset by training InPHYNet on different permutations of grade-wise data.

2. Related work

We position our work in the literature by focusing on three key areas related to our work: (1) educational technologies for interactive learning systems, (2) multitask learning, and (3) text classification.

2.1. Educational technologies for interactive learning systems

The field of education is one of the most fundamental spheres of facilitating learning and acquiring knowledge. Research in the confluent space of technology and education has gathered a lot of interest in recent years. Some of this work has focused on free-text or subjective question–answers of students, and their evaluation. Noorbehbahani and Kardan [13] provided an algorithm for the assessment of free-text answers. Westera et al. [14] proposed techniques for the automated scoring of students' essays. Both of these helped reduce the workload of teachers in terms of evaluation. Rodrigues and Oliveira [15] presented a system that is based on the free-text answers of students and is capable of monitoring their progress as well as provides a formative assessment of the students. Atapattu et al. [16] indicated that course material, delivered as graphs and visuals, help students better because they can relate to concepts easily. They also provided an automated method for generating concept maps based on lecture slides.

There has been an increasing focus on building end-to-end QA based interactive dialog systems for facilitating better learning. Agarwal et al. [17] proposed a model framework for building a QA system for any subject/course material. Afzal et al. [18] discussed the prototype of a dialog-based intelligent tutoring system, called the Watson Tutor. However, both these models are limited in providing a smooth human-like conversational interaction. A lot of work has been pivoted on the Student Response Analysis (SRA) part of a dialog system. SRA is used in systems that evaluate student answers. Dhamecha et al. [19] noted the problem of lack of an adequately labeled dataset for SRA and addressed it via a cost-effective method of data collection. Marvaniya et al. [20] handled the evaluation part of SRA and proposed the designing of evaluation rubric using a model captured from a focused set of responses.

We contribute towards building a QA system for physics school education in the near future. We propose a novel deep learning architecture, namely, InPHYNet that is capable of assigning a set of multiple labels to a given text. This architecture can also be used for question-type annotation and can be positioned directly within a QA system pipeline that would lead to improvements in the quality of the system. We also introduce an annotated physics dataset for multi-label text classification that can be used for several tasks, such as question-type annotation, answer-type classification, and topic clustering.

2.2. Multitask learning

Multitask learning (MTL) is a machine learning technique in which multiple learning tasks are solved simultaneously by exploiting similarities across tasks [21]. This proves extremely helpful in improving the efficiency and accuracy of the task-specific models. Multitask learning typically consists of a primary task and several auxiliary tasks that are trained simultaneously by using a shared set of hidden layers in the MTL architecture. By utilizing the entire data from both the primary and auxiliary tasks, features learned for representing the common data attributes become highly discriminative and reinforced. This helps in better training of all the tasks together and hence, improves the accuracy of these tasks. Furthermore, since all the tasks are trained simultaneously, the training latency improves by a large factor in comparison to training different tasks asynchronously. Hence, multi-task learning helps in enhancing the quality of results obtained at an expedited rate.

The most common form of MTL entails the co-training of related tasks [22]. These tasks share proximate input representations and subsequently use separate networks adapted to each task. All the tasks are jointly co-trained together with a number of shared hidden representation layers. MTL has been found to be very useful in supervised text classification problems [23–25] as well as in semi-supervised problems [26]. A number of such MTL techniques are presented by Zhang and Yang [27].

Recently, with the advances in sequence to sequence models, attention [28] has become a very popular mechanism to align sequences of data. MTL models have also greatly benefited from using formulations of attention within their model architectures. Lan et al. [29] presented an MTL attention-based model to address implicit discourse relationship representations. They made use of a sigmoid-gated sharing strategy for training

[2] Code available on request.

their multi-task framework for the tasks of learning knowledge from annotated and unannotated corpora. Liu et al. [30] leveraged both MTL and attention in the computer vision domain by learning task-specific feature-level attention using a single shared MTL network. This architecture allowed for the learning of task-specific global features while simultaneously allowing sharing of visual features across a set of diverse tasks. Stickland and Murray [31] introduced 'projected attention layers' to be used along with the multi-task training to ensure high quality adaptation of the BERT [32] model's sentence representation quality. MTL and attention models have also been explored in the speech domain. Zhang et al. [33] explored leveraging attention mechanisms embedded within MTL LSTM-based acoustic models for distant speech recognition. Their experiments clearly demonstrate that their model improves robustness for their primary senone classification task and auxiliary feature enhancement task.

Gupta et al. [34] recently proposed a novel MTL network architecture, called GIRNet, that learns to derive task-based composite state sequences. GIRNet architecture makes use of the fundamental MTL assumption that the number of auxiliary tasks' training instances will always be much greater than the primary task's training instances due to the scarcity of primary task data. GIRNet uses recurrent neural networks with Long Short Term Memory (LSTM) blocks as the base neural network architecture. It consists of a primary LSTM network and auxiliary LSTM networks. The auxiliary task's training instances are passed through the auxiliary LSTMs for training. However, each primary task's training instance is passed through the primary LSTM network as well as the auxiliary LSTMs for training. For each primary task's training instance, the auxiliary LSTM network states are combined by performing a gating operation. These are passed-on to the primary LSTM network. This creates a robust mechanism for simultaneous training of the primary and auxiliary networks.

In this work, we propose an improvement on the GIRNet model by applying a weight alignment layer to the MTL network that is capable of judging the importance of each auxiliary task's LSTM network to the primary input data. This weight alignment approach helps in enhancing the representational capacity of the primary LSTM network's state sequences by assigning a weight to each auxiliary LSTM network in association with each primary task training instance. We call this improved network model as InPHYNet because it is trained on physics subject and is used for multi-label physics text classification.

2.3. Text classification

Text classification is one of the very fundamental exercises in Natural Language Processing (NLP) which helps us classify data (structured/unstructured) into various categories based on its content. The problem of text classification has been popular since the late '90s with advancements in results every year. Initial methods approached this with Bayesian classification techniques [35] and the unigram language model [36–38]. With growing popularity and importance, researchers started using advanced machine learning techniques such as Support Vector Machine (SVM) [39]. SVM has also been applied in the field of social media text classification to improve information filtering [40].

With the increasing demand and enormous applications of text classification, deep learning methods such as Convolutional Neural Networks (CNN) [41], Recurrent CNN [42], long short term memory (LSTM) [43] have been used to classify various kinds of data. Likewise, there is a regional text classification approach using an attention mechanism [44] in addition to the deep learning model framework. Furthermore, recent developments in NLP techniques have improved the ability to represent textual data in well-structured mathematical formulations [32,45–49].

Table 1
Description of nine labels identified for the CBSE Physics school curriculum.

Label type	Description
Definition	What is ... ?, Define ..?
Causes	What causes ... ?, What leads to ... ?, How is...?
Examples	Give some examples of ...?
Reasoning	Explain the working of ...?, Give reasons why ...?
Property	What are the attributes of ... ?
Types	What are the different types of ?
Formula	Write down the formula for ... ?, How is ... calculated?
Effects	What are the effects of ... On ... ?, What happens when ...?
Relation	How is ... related to ... ?, How is ... different from ...?

Multi-label text classification entails the assignment of one or more labels to each input data (paragraph) [50]. These tasks are often considered to be more challenging as compared to binary/ multi-class text classification problems because it requires the assignment of labels to each input paragraph to be variable (one or more). Recently, there has been a lot of work done at the cross section of deep learning and multi-label classification. Pereira et al. [51] conducted a comprehensive study on the feature extraction techniques which are germane to multi-label classification models. Ahmed et al. [52] conducted robust experimentation with classic machine learning models and problem transformation techniques to convert multi-label classification into a single multi-class classification problem. Chang et al. [53] proposed a deep learning framework that takes inspiration from approaches used in information retrieval. It solves the problem by using a three pronged approach involving label indexing, matching, and ranking. Other methods include using a neural network for the probabilistic scoring of labels [54,55] and a label predictor to identify the best relevant and irrelevant labels [56].

Our approach differs from all the above approaches. We apply an attention mechanism to an interleaved recurrent network that suits our goals for multi-label text classification via multitask learning framework.

3. Overview of datasets

We use two datasets in this paper to conduct our experiments and analysis. The first dataset, "Multi-label Physics K-12 dataset", is a novel dataset that has been created by us to help promulgate the research in the area of educational question-answering. The second dataset is publicly available, namely, "Experimental Data for Question Classification" released by Li and Roth [57]. We describe both these datasets below.

3.1. Multi-label physics K-12 dataset

3.1.1. Dataset creation

For creating the dataset, we focus on grade 6th to 12th physics as taught in the federally supported CBSE curriculum in India. Corresponding to each chapter of every grade, we collect chapter notes from the free public websites that provide help to students in CBSE curriculum. Each of these chapter notes is further composed of paragraphs. In order to correctly capture the context specified in each of these paragraphs, we identify nine label types.

These labels are required to satisfy the below two conditions:

- The identified labels should capture all possible contextual information exhaustively. For example, if a paragraph contains a definition followed by an example, then the labels should capture both the *definition* and *example* information aptly.
- They should be independent of each other. This implies that each label should capture a unique contextual information about the paragraph. Therefore, the labels should

Table 2
Examples of paragraphs and their corresponding labels in the dataset.

Example paragraph	Labels	Possible questions
Force can make a stationary body in motion. For example a football can be set to move by kicking it, i.e. by applying a force.	*Examples*, *Effects*	What are the effects of Force? Describe a property of Force? Give an example of the effects of Force.
When the mirror is a part of a sphere, it is called spherical mirror. Spherical mirrors are of two types. In concave mirror, the reflective surface is inside the sphere, i.e. is depressed. In convex mirror, the reflective surface is outside the sphere, i.e. is bulged or protruded.	*Definition*, *Types*	What is a spherical mirror? What are types of Spherical mirrors? Explain concave and convex mirror.

not be overlapping with each other in capturing information context. For example, if a paragraph contains a definition followed by a formula and an example, we essentially have three unique information entities in the paragraph - *definition*, *formula*, and *example*.

We conducted extensive annotation to understand the possible label types that could encode the contextual information present in each paragraph, while satisfying the two conditions stated above. Based on the annotation of CBSE physics chapters, we could identify nine unique label types that are independent of each other (in capturing contextual information) and could also exhaustively encapsulate all different types of information contexts present in paragraphs. These nine label types are described in Table 1.

Each paragraph can, therefore, be annotated by either one or multiple labels that we term as the *labelset* for each paragraph. The labelset for each paragraph is a subset of the above nine label types. Some examples from the dataset are shown in Table 2. In Fig. 1, we visually present a few preliminary observations of the dataset.

3.1.2. Dataset statistics

Our dataset is multi-label in nature, *i.e.*, every paragraph (training sample) in the dataset can have one or more labels. For instance, consider the following paragraph:

"*When an object repeats its motion after a fixed interval of time, it is said to be undergoing periodic motion, say, for example, pendulum.*"

We note that the above paragraph is associated with two labels – *Definition* and *Examples*. We, therefore, scrutinize the dataset and quantify the distribution of data points and labels by calculating three major statistical measures. Let us define N to be the total number of training samples (paragraphs) in the dataset, L to be the total number of labels that are possible for a training sample and $y_j^{(i)}$ to be the binary value of the jth label for the ith training sample. We measure the following:

- Label Cardinality (LC): It is a measure of the average number of labels per training sample.

$$LC = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L} y_j^{(i)}$$

- Label Density (LD): It is a measure of the average number of labels per training sample divided by the total number of labels.

$$LD = \frac{1}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L} y_j^{(i)}$$

- Diversity (d): It is a measure of the average number of labels per training sample multiplied by the total number of training samples. In simpler terms, it is the total number of labels over all training samples.

$$d = \sum_{i=1}^{N}\sum_{j=1}^{L} y_j^{(i)}$$

Table 3
Dataset statistics.

Statistic	Value
Number of labels (L)	9
Number of training samples (N)	4199
Label cardinality (LC)	1.68706835
Label density (LD)	0.1874520388
Diversity (d)	7084
Distinct labelsets	133
Most frequent labelset	*Definition*, 975

The dataset statistics are listed in Table 3.

3.1.3. t-SNE visualization

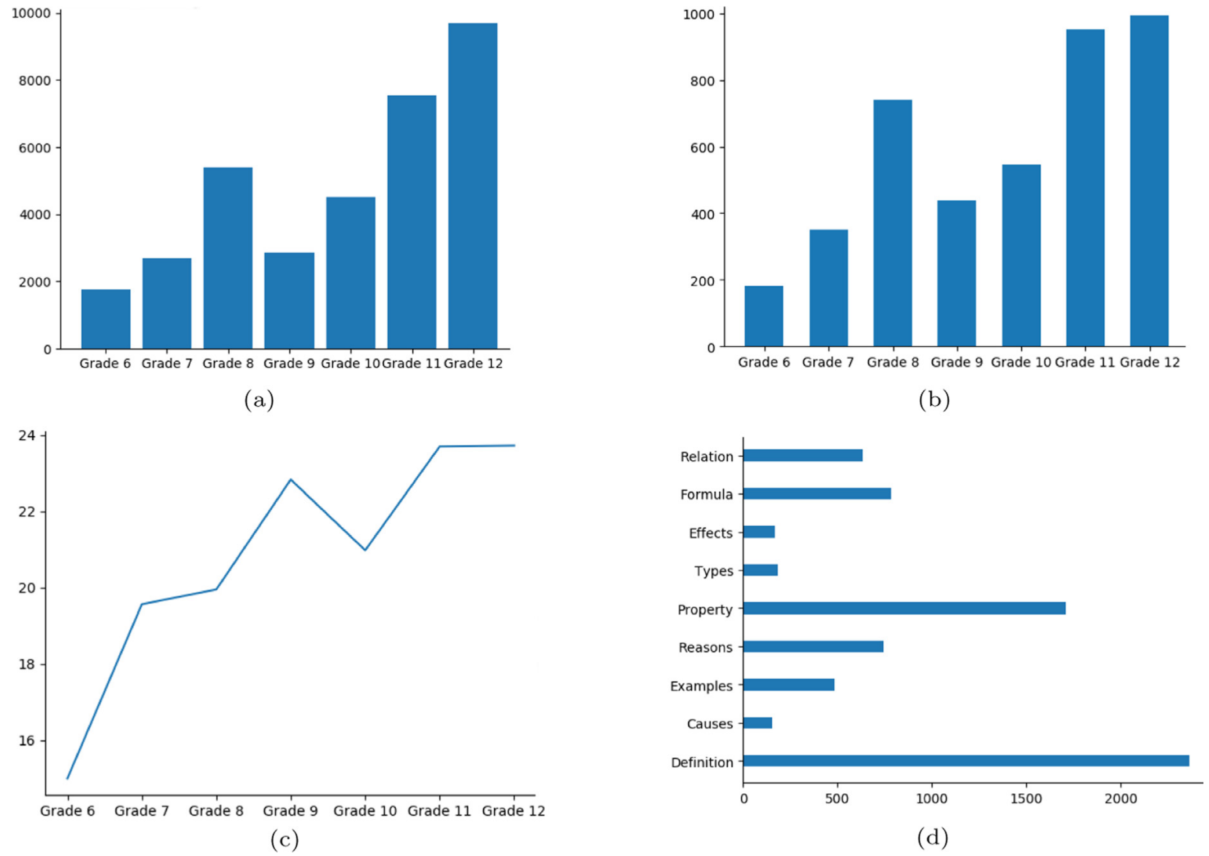
In order to visualize the distribution of data across grades, we performed t-SNE feature reduction and reduced the data to two dimensions. The t-SNE plot for the same is visualized in Fig. 2. Each paragraph in the dataset is labeled with the grade, it belongs to. There are six grades and each grade is represented by a different color in the plot. The graph depicts high correlation among the concepts across various grades. Grade 12 data is distributed across the whole graph because it covers all the concepts from 6th to 11th grades in greater detail. It is evident from the plot that separating data on the basis of a specific grade is a very challenging task.

3.1.4. Wordcloud

Wordcloud is used to visually interpret large scale text data. The size of a word in the wordcloud is proportional to its frequency in the text dataset. The wordcloud for our dataset is shown in Fig. 2. This has been plotted after removing stop words from the dataset corpus. The coloring has been done to make it visually more appealing and readable. We observe that various technical terms such as *magnetic field*, *electric current* and *velocity* have significant size denoting higher frequency across the dataset. Thus, this plot can be used to infer the topics that are taught most and have maximum importance.

3.1.5. Lexical dispersion plot

We generated a lexical dispersion plot (Fig. 3) for 14 common technical terms that were drawn from the dataset corpus. These terms were widely distributed across the data in the 6th to 12th physics curriculum. The motivation behind analyzing this lexical dispersion plot were: (i) to provide an alternative means of visualizing the prevalence of these technical terms in the dataset, (ii) to verify the presence of a clustering pattern (i.e., whether a term featured heavily for a particular grade or for a set of grades, or whether it was widely spread across all grades), and (iii) to



(a) (b) (c) (d)

Fig. 1. (a) Vocabulary sizes over different grades, (b) Number of paragraphs in each grade, (c) Number of paragraphs per document in different grades, (d) Number of paragraphs per label type.

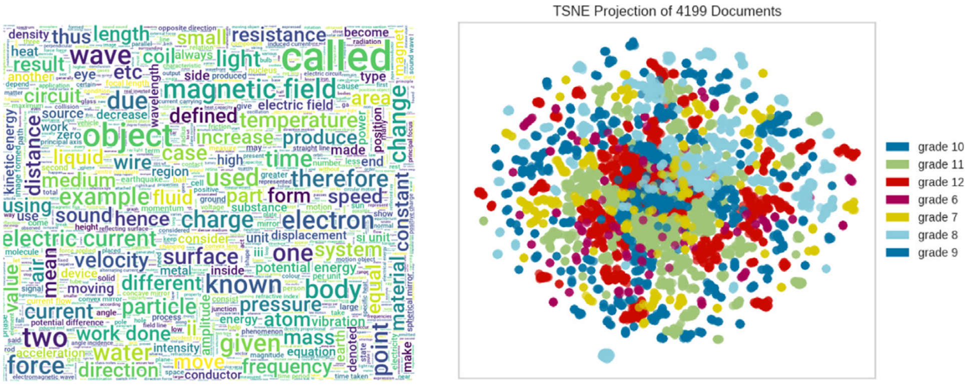


Fig. 2. Left: Wordcloud for the dataset. Right: *t*-SNE visualization of the data points categorized by grades.

analyze any correlation between these technical terms and the grades.

It is observed that terms like *Force* and *Energy*, taught in almost every grade, appear throughout the plot. On the other hand, terms like *semiconductor*, *wave*, and *atom* appear mostly in the higher grades because they are taught at a later stage. This helps immensely in understanding how the concepts are taught to the students and can be used to suggest improvements in the educational system.

3.2. Experimental data for question classification

We now describe the second dataset that was used in the experiments. This dataset is a publicly available dataset, released by researchers from The University of Illinois at Urbana–Champaign (UIUC), USA [57] to support research in the field of building automated question-answering systems. It is primarily used for learning an efficient question-type classifier for a given input question. This dataset contains around 5500 labeled questions with a hierarchical labeling structure. There are six parent classes such as *Entity*, *Description*, etc. Each class has various subclasses ranging from 2 to 22 in number. Brief dataset statistics are tabulated in Table 4. Some sample questions and their corresponding labels with meanings are shown in Table 5.

4. InPHYNet: Proposed architecture

In this section, we propose a general multi-task learning framework that can be used widely for improving the performance of the primary task by leveraging the network capabilities of the auxiliary task in a coherent manner.

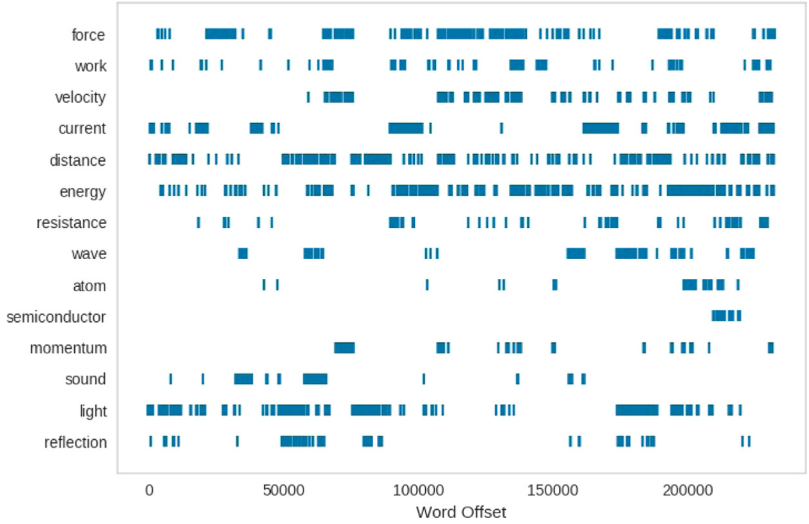


Fig. 3. Lexical dispersion plot.

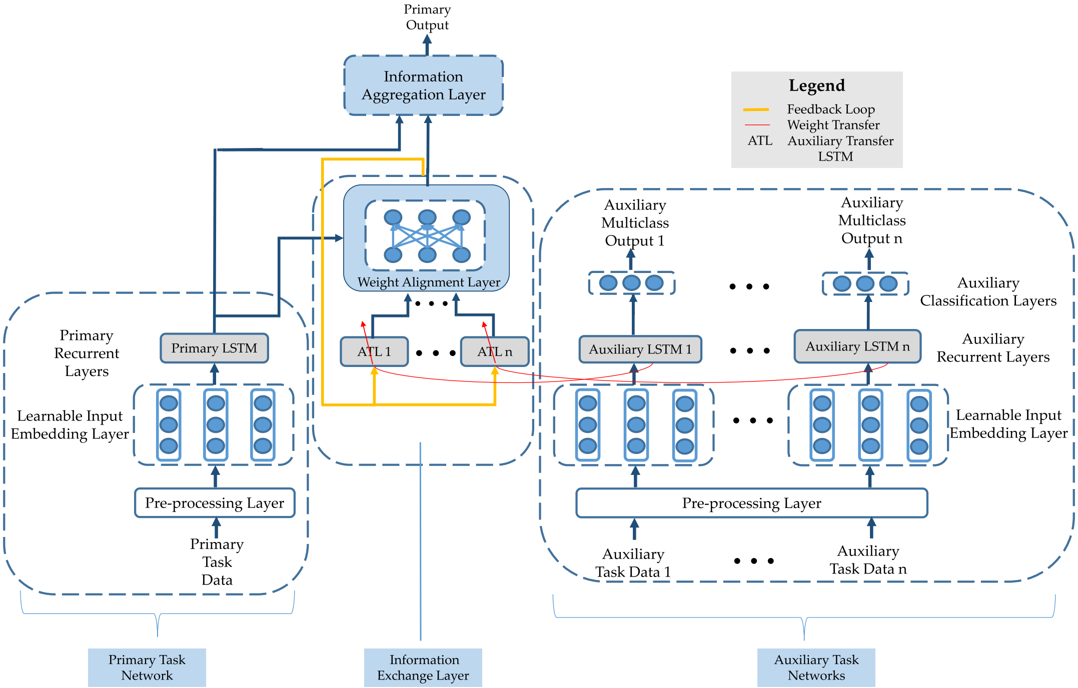


Fig. 4. A general structure of the InPHYNet framework. The primary task network takes in a primary task input and processes it within the primary LSTM network. Next, the output of the primary LSTM network is passed into the weight alignment layer present inside the information exchange layer. Similarly, n auxiliary task networks take in auxiliary inputs and process them within the auxiliary LSTM networks. In every training epoch, the weights of the auxiliary LSTM networks are transferred to the Auxiliary Transfer LSTM (ATL) networks (depicted by the red arrows). There is a continuous feedback-loop mechanism in which the information exchange layer works (depicted by the yellow arrow). The individual auxiliary task output predictions are done by the auxiliary task classification layers. The primary task output prediction is done by the information aggregation layer that takes inputs from the primary LSTM network and the information exchange layer.

Table 4
Statistics of experimental data.

Statistic	Value
Number of labels	47
Number of training samples	5452
Most frequent label	*ind*, 962

Table 5
Sample questions and labels from experimental data.

Question	Label
Who was The Pride of the Yankees?	*ind* - an individual
What causes asthma?	*reason*- reasons
What year was the NAACP founded?	*date*- dates

Our problem setting consists of one primary task and n auxiliary tasks. Usually, the labeled data for the primary task is small in size, whereas the labeled data for the auxiliary tasks is in abundance. We aim to leverage the large amount of auxiliary task data to improve the performance of the primary task. This can be achieved by seeking the most relevant sequences of auxiliary data for a given primary data sample. We identified the most important auxiliary data sequences by using an attention-based

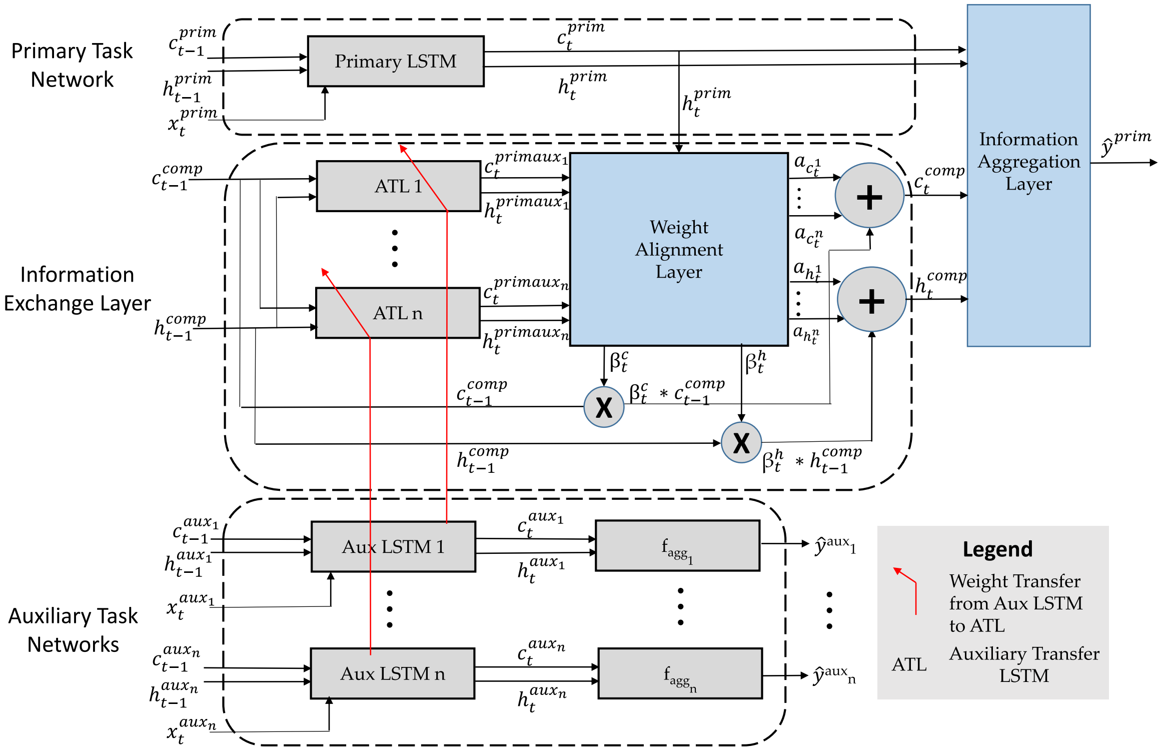


Fig. 5. A cross-sectional view of the InPHYNet architecture. c_{t-1}^{prim} and h_{t-1}^{prim} are the primary cell and hidden states produced by the primary LSTM network at time step $t-1$. x_t^{prim} is the primary input at time step t. c_{t-1}^{comp} and h_{t-1}^{comp} are the output composite cell and hidden states produced by the information exchange layer at time step $t-1$. These output composite states are fed back into the information exchange layer at time step t, forming the continuous feedback-loop mechanism depicted by the yellow arrow in Fig. 4. $c_t^{primaux_i}$ and $h_t^{primaux_i}$ are the intermediate cell and hidden composite states, respectively, produced by the ith ATL network. These intermediate composite states are provided as inputs to the weight alignment layer. $c_{t-1}^{aux_i}$ and $h_{t-1}^{aux_i}$ are the auxiliary cell and hidden states, respectively, produced by the ith auxiliary LSTM network. $x_t^{aux_i}$ is the auxiliary input at time step t to the ith auxiliary LSTM network. The red arrows depict the transfer of weights from the auxiliary LSTM networks to the ATL networks. ⊕ represents element wise addition of two vectors.

mechanism that quantifies the pertinence of each auxiliary task data sample with respect to a primary data sample.

To this end, we modified a recently proposed Multi-Task Learning (MTL) framework, called GIRNet [34], by adding a novel weight alignment layer using a weight transfer strategy and including Auxiliary Transfer LSTM (ATL) networks. Our multi-task learning model called InPHYNet learns to selectively attend to auxiliary recurrent neural networks (RNNs) in the model based on their importance to each primary data sample. We used LSTMs as our RNN blocks. However, these can be replaced by other RNN blocks such as Gated Recurrent Units (GRUs) [58] or Neural Turing Machines (NTMs) [59].

We briefly explain the general working of InPHYNet here and defer the technical details of the individual components to the subsequent sections. The general architecture of InPHYNet is shown in Fig. 4. This framework consists of a primary LSTM network and n auxiliary LSTM networks. The primary input and the auxiliary inputs are passed into a pre-processing layer which is responsible for converting raw textual inputs into a vectorized form. The pre-processed primary and auxiliary vectors are passed into individual learnable input embedding layers. The learnable input embedding layers can be used to perform dimensionality reduction on the pre-processed vectors.

Once we obtain the vector outputs from the learnable input embedding layers, the primary embedding layer vector is passed into the primary LSTM network, whereas the auxiliary embedding layer vectors are passed into their corresponding auxiliary LSTM networks.

Next, we introduce an *information exchange layer* that is responsible for transferring and combining the information from the auxiliary LSTM networks in a parameterized fashion. The information exchange layer is run in a feedback-loop based fashion, wherein the output of the layer at a particular time instance t is the input to the layer at the next time instance $t+1$. The information exchange layer consists of n auxiliary transfer LSTM (ATL) networks and a weight alignment layer. ATL networks share weights with the auxiliary LSTM networks and are used to produce 'information vectors' (which we call as composite states) that are passed into the weight alignment layer.

The *weight alignment layer* combines the n composite states from the ATL networks to produce a combined information sharing representation of the composite states. The primary objective of the weight alignment layer is to quantify the significance of each ATL network to the primary data sample by assigning weights to each of the composite states produced by the individual ATL networks. Once equipped with weights for each individual composite state, we form weighted composite states. These weighted composite states are next aggregated in the weight alignment layer to form output composite states.

These output composite states are passed into the *information aggregation layer* along with the primary LSTM network's output states. The information aggregation layer is responsible for combining the information present in the output composite states and the primary LSTM network's output states to produce the primary task output.

The auxiliary task outputs are produced by passing the auxiliary LSTM networks' output states into separate aggregator functions for each individual auxiliary task. An example of an auxiliary task aggregator function can be a softmax layer which produces a normalized probability distribution for each class in an auxiliary task.

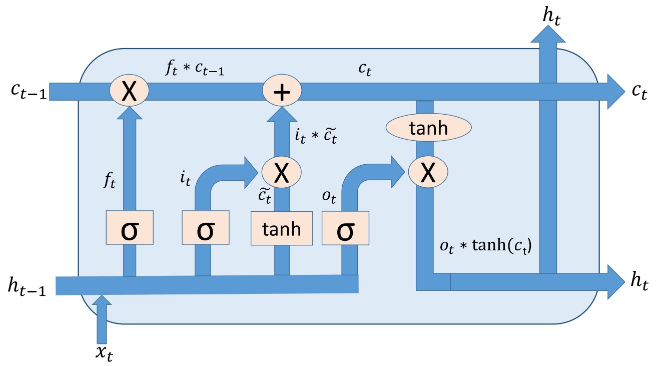


Fig. 6. LSTM block diagram.

A cross sectional view of the InPHYNet framework with the individual state representations is shown in Fig. 5. Because InPHYNet is primarily based on LSTM networks, we now provide a brief explanation of an LSTM network in Section 4.1. Next, we present details on each sub-component of the InPHYNet framework.

4.1. LSTMs and pre-processing layers

Since InPHYNet uses LSTM networks extensively, we follow the below stated convention to describe an LSTM network in the further Sections. At time t, for an input x_t, hidden state h_{t-1} and cell state c_{t-1}, we represent the output hidden state as h_t and output cell state as c_t. We use the following equation to represent the LSTM network (see Fig. 6).

$$h_t, c_t = LSTM(x_t, h_{t-1}, c_{t-1})$$

The input data for both the primary and auxiliary tasks is in the form of raw text. We need a vectorized representation of this data so as to map the input sentences into vectors which can be fed into our InPHYNet architecture. This conversion of the raw input sentences into vectors can be done by using multiple preprocessing techniques such as:

- **TF–IDF Vectors**: Term Frequency–Inverse Document Frequency (TF–IDF) calculates a value/score for each word which represents it's relative importance in the entire text. This score is based on 'normalized term frequency' and 'inverse document frequency'. It is computed as:

  $$w_{i,j} = tf_{i,j} \cdot log\left(\frac{N}{df_i}\right),$$

  where i is the input sentence index, j is the vocabulary term index, N is the total number of documents (can be input sentences or paragraphs), $tf_{i,j}$ is the term frequency of the jth term in the ith sentence, and df_i is the document frequency of the ith term. It has shown very good results in determining the relevance of a word in documents [60].
- **Count Vectors**: In this technique, a unique vocabulary is extracted from the entire text corpus. The vector representation for each input sentence is encoded as the counts of each vocabulary term present in that corresponding sentence.
- **Doc2Vec** [61]: This technique converts the input sentence into a vector by learning to predict a word based on the surrounding contextual information. These vectors are a set of numbers that help the network in understanding the semantics and learning to perform the desired task.
- **Elmo word vectors** [46]: This technique generates vectors for each word by using context specific information (semantic and syntactic) after training a deep bidirectional language model.
- **BERT word vectors** [32]: This technique generates vectors for each word by training a transformer network to extract deep bidirectional representations.
- **MT-DNN word vectors** [47]: It generalizes the BERT bidirectional language model by applying an effective regularization mechanism and creates the word vectors.

4.2. Learnable input embedding layer

Once the input raw text is converted into vector representations using one of the preprocessing techniques mentioned above, these vectors can be fed into the InPHYNet architecture. However, these vectors are static vectors and are not trained along with the rest of the architecture. This can be a hindrance because the input representations are very high dimensional and can be sparse. Therefore, a mechanism is needed that can simultaneously perform dimensionality reduction and preserve the context similarities between the words in the reduced multi-dimensional space.

Therefore, we use an input embedding layer to represent these preprocessed sentence vectors. This layer is trained with the rest of the InPHYNet network. There are separate learnable input embedding layers for the primary task and the different auxiliary tasks.

4.3. Auxiliary LSTM networks

Separate auxiliary LSTM networks are used for each individual auxiliary task. Let us assume that there are n auxiliary tasks and n auxiliary LSTM networks. We denote $x^{aux_j} = \left[x_1^{aux_j}, x_2^{aux_j}, \ldots, x_n^{aux_j}\right]$ to be an auxiliary task input for task j, where $x_i^{aux_j}$ is the ith time training sample. We also denote y^{aux_j} to be the corresponding training label for task j. We denote the jth auxiliary LSTM network's hidden state at time instance t as $h_t^{aux_j}$. Similarly, the jth auxiliary LSTM network's cell state at time instance t is denoted as $c_t^{aux_j}$. By this convention, the jth auxiliary LSTM network equation is:

$$h_t^{aux_j}, c_t^{aux_j} = LSTM(x_t^{aux_j}, h_{t-1}^{aux_j}, c_{t-1}^{aux_j}) \qquad (1)$$

The final output prediction for each separate auxiliary task j is performed by a separate aggregator function (typically a neural network) f_{agg_j} that aggregates these auxiliary task output states as:

$$\widehat{y}^{aux_j} = f_{agg_j}(h_1^{aux_j}, \ldots, h_n^{aux_j}). \qquad (2)$$

The loss for each auxiliary task is computed as

$$L_{aux_j} = loss(y^{aux_j}, \widehat{y}^{aux_j}), \qquad (3)$$

where function such as L1-norm, L2-norm, or cross entropy can be used as *loss* functions. The loss of each separate auxiliary task is optimized separately, but added together to the final network loss.

4.4. Primary LSTM network

We denote $x^{prim} = \left[x_1^{prim}, x_2^{prim}, \ldots, x_n^{prim}\right]$ to be a primary task input, where x_i^{prim} is the ith time training instance. We also denote y^{prim} to be the corresponding training label. For a multi-label setting, our model assumes that the labels are pairwise independent. Hence, theoretically there is no limit on the number of different multi-label classes that can be supported.

We denote the primary LSTM network's hidden state at time instance t as h_t^{prim}. Similarly, the primary LSTM network's cell state at time instance t is denoted as c_t^{prim}.

Then, by our convention, the primary LSTM network equation is:

$$h_t^{prim}, c_t^{prim} = LSTM(x_t^{prim}, h_{t-1}^{prim}, c_{t-1}^{prim}) \tag{4}$$

The primary hidden state h_t^{prim} is further used in the weight alignment layer to assign weights to the composite states. These weighted composite states are aggregated in the weight alignment layer to produce output composite states. The output composite states are used for primary task prediction along with the primary LSTM network's output cell and hidden states.

4.5. Information exchange layer

We use the information exchange layer, to efficiently leverage the learning done by the auxiliary LSTM networks, with regard to the primary task data samples. The information exchange layer consists of two components: (*i*) n auxiliary transfer LSTM (ATL) networks , and (*ii*) a weight alignment layer. The entire information exchange layer works in a feedback-loop arrangement where the output in the tth time instance is the input to the layer in the $(t+1)$th time instance. Next, we explain both the components of the information exchange layer in depth.

4.5.1. Auxiliary Transfer LSTM (ATL) networks

The ATL networks are used to provide 'information vectors' or the composite states to the weight alignment layer. We use a weight transfer strategy for the ATL networks wherein at every time instance t, we transfer the weights of the jth auxiliary LSTM network to the jth ATL network. This ensures that we retain the information context of the auxiliary LSTM networks in the ATL networks.

Our ATL networks differ from a standard LSTM network because they do not take any input. Hence, the transfer strategy used for the ATL networks helps in processing the cell and hidden states in the absence of any input.

Our ATL network input cell and hidden states at a time instance t are the output cell and hidden composite states that are produced as the outputs of the weight alignment layer at time instance $t-1$. This ensures that the feedback-loop arrangement for the information exchange layer is satisfied.

We denote the output hidden composite state at time t as h_t^{comp}, the output cell composite state at time t as c_t^{comp}, the jth ATL network's hidden composite state at time t as $h_t^{primaux_j}$, and the jth ATL network's cell composite state at time t as $c_t^{primaux_j}$. Thus, we represent the jth ATL network functionality as:

$$h_t^{primaux_j}, c_t^{primaux_j} = LSTM(h_{t-1}^{comp}, c_{t-1}^{comp}) \tag{5}$$

The composite states that are produced by the ATL networks are passed into the weight alignment layer.

4.5.2. Weight alignment layer

There are two major input components to the weight alignment layer:

- the composite states that are outputs of the ATL networks and
- the hidden state of the primary LSTM network.

The major goal of the weight alignment layer is to assign weights to each of the composite states produced by the ATL network to factor in the relevance of each composite state to the current primary data sample. This is achieved by introducing an attention-based mechanism [62] to improve the auxiliary context sensitization of the network at every primary input data sample. The composite states (hidden states and cell states) are converted into the weighted alignment cell vectors $a_{c_t}^{primaux_j}$ and weighted alignment hidden vectors $a_{h_t}^{primaux_j}$ where j represents each ATL network used, t is the current time instance, α_{c_j} is the weight for the alignment cell vector and α_{h_j} is the weight for the alignment hidden vector. The weight alignment layer helps in quantifying the relative importance of the composite states to the current primary data sample. This importance is factored in by using the primary task hidden state h_t in the weight alignment layer.

The weight alignment layer is a neural network with some non-linear layers followed by a sigmoid activation layer. The sigmoid activation layer ensures that all the alignment vector weights are squashed within the range [0, 1]. This normalizes the alignment vectors and ensures that the individual alignment vector elements do not overflow over a certain threshold. Further, this weight alignment layer network is used to produce the weights for the alignment vectors. The alignment cell and hidden vectors and their weights can be represented as:

$$\alpha_{c_j} = \sigma\left(f_c\left(c_t^{primaux_j}, h_t^{prim}\right)\right), \tag{6}$$

$$\alpha_{h_j} = \sigma\left(f_h\left(h_t^{primaux_j}, h_t^{prim}\right)\right), \tag{7}$$

$$a_{c_t^j} = \alpha_{c_j} c_t^{primaux_j}, \tag{8}$$

and

$$a_{h_t^j} = \alpha_{h_j} h_t^{primaux_j}, \tag{9}$$

where f_c and f_h are the non-linearity functions used in the weight alignment layer and σ is the sigmoid activation function.

These alignment vectors are used to compute two context vectors for that particular primary task time instance, *i.e.*, the hidden context vector β_t^h and the cell context vector β_t^c. These context vectors help to capture the net importance of all ATL networks for the primary task data sample for the time instance t. They are represented as:

$$\beta_t^h = \frac{\sum_{j=1}^m a_{h_t^j}}{\sum_{j=1}^m \alpha_{h_j}}, \tag{10}$$

and

$$\beta_t^c = \frac{\sum_{j=1}^m a_{c_t^j}}{\sum_{j=1}^m \alpha_{c_j}}. \tag{11}$$

The output hidden composite state h_t^{comp} and output cell composite state c_t^{comp} are computed by taking into account the weighted alignment vectors as well as the context vectors. Therefore, these are represented as:

$$h_t^{comp} = \sum_{j=1}^m a_{h_t^j} + \beta_t^h \cdot h_{t-1}^{comp}, \tag{12}$$

and

$$c_t^{comp} = \sum_{j=1}^m a_{c_t^j} + \beta_t^c \cdot c_{t-1}^{comp}. \tag{13}$$

Thus, at every time instance t, the primary task feature representation is enhanced by employing a weighted alignment model that will learn to take into account the relative importance of each composite state produced by the ATL networks in that particular time step. These output composite states are used to update the composite states in the next step, creating a continuous feedback loop that greatly enhances the representation of the primary task output state information.

4.6. Information aggregation layer

This layer is used to produce the final primary task output. This final output for the primary task is a conflated representation of both the output composite states (from the weight alignment layer) as well as the primary task output states (from the primary LSTM network). The prediction is performed by a neural network g_{agg} which aggregates the output composite states and the primary task output states:

$$\widehat{y}^{prim} = g_{agg}([h_1^{prim}, \dots, h_n^{prim}]; [h_1^{comp}, \dots, h_n^{comp}]) \tag{14}$$

The loss for the primary task is computed as:

$$L_{prim} = loss(y^{prim}, \widehat{y}^{prim}) \tag{15}$$

where *loss* can be any loss function such as L1-norm, L2-norm, or the cross entropy.

4.7. Joint optimization of primary and auxiliary losses

The primary LSTM network, auxiliary LSTM networks and the information exchange layer are jointly optimized by simultaneous training. For every iteration, we take a sample of primary input x^{prim} and one sample each for every auxiliary task input x^{aux_j} where j is the auxiliary task index. The primary (L_{prim}) and auxiliary (L_{aux_j}) losses are computed as described above. The final loss of the overall network is computed as a weighted sum over the primary and auxiliary losses with a parameterized weight of μ_j given to each of the auxiliary task losses:

$$L_{net} = L_{prim} + \sum_{j=1}^{n} \mu_j L_{aux_j} \tag{16}$$

5. Experimental setup

5.1. Baseline models

In order to compare the performance of the proposed InPHYNet architecture, we consider 7 baseline machine learning models:

- **Gaussian Naive Bayes classifier (Gaussian NB):** It is a probabilistic classifier that is based on applying Bayes' theorem with strong independence assumptions between the features. Here the assumption is that the input data points are drawn from a gaussian distribution.
- **Multi-label K-nearest neighbors (ML-KNN)** [63]**:** It is a classifier that employs a lazy multi-label approach on top of the existing K-nearest neighbors algorithm. After identifying the K-nearest neighbors for a test point, maximum a posteriori (MAP) principle is utilized to determine the labelset for the corresponding test point.
- **Decision Tree classifier:** It is a predictive modeling approach to target classification based on the relative importance of the input features.
- **Random Forest classifier:** It is an ensemble learning technique where a multitude of decision tree classifiers are clubbed together to improve target classification accuracy.
- **Multinomial Naive Bayes classifier (MultinomialNB):** It is a probabilistic classifier similar to a Gaussian naive Bayes classifier with the assumption that the input samples are drawn from a multinomial distribution.
- **Multi Layer Perceptron with relu (MLP(relu, 1, 100)):** A multi layer perceptron is a feed-forward neural network. This particular neural network consists of one hidden layer with 100 neurons, each having a relu activation function. The relu function is defined as $relu(x) = max(0, x)$.
- **Multi Layer Perceptron with sigmoid (MLP(sigmoid, 1, 100)):** This particular perceptron network consists of one hidden layer with 100 neurons each having a sigmoid (σ) activation function. The sigmoid function is defined as $\sigma(x) = \frac{1}{1+e^{-x}}$.

Along with the above mentioned machine learning models, we train three deep learning models, namely a vanilla LSTM classifier, the GIRNet model and the pre-trained BERT model. The hyperparameters for these models are stated below:

- **Vanilla LSTM classifier:** We use a classifier with an LSTM network block followed by a softmax activation layer. The LSTM network block uses 512 dimensional hidden and cell states.
- **GIRNet model:** We use the GIRNet model with one auxiliary task. The primary and auxiliary datasets we use to train the GIRNet are used in a similar setting as described for training InPHYNet in Section 5.2.
- **Pre-trained BERT model:** We use a pre-trained BERT model containing 12 transformer layers with 768 hidden units, 12-multi-attention heads, and 110M parameters.[3] This model was pre-trained on the BookCorpus[4] and English Wikipedia[5] datasets. We fine-tune the final classification layer of this model for our primary multi-label classification task.

All the aforementioned classifiers can be used directly for binary or multiclass classification settings. However, since our problem setting is multi-label in nature, we apply three pertinent problem transformation techniques to each of these classifiers (except ML-KNN) to make them suitable for performing multi-label classification. The following problem transformation techniques were used in this work:

- **Binary Relevance**: Let us suppose there are C different classes for the multi-label classification problem. In the binary relevance technique, each of the C labels are treated as independent target labels and therefore C independent classifiers are trained for a binary classification setting. For a test sample, each classifier predicts an output for its own target label.
- **Classifier Chain** [64]: This problem transformation technique involves converting a multi-label classification problem into several binary classification problems. Here, labels are predicted sequentially for each classifier, and the output of all previous classifiers are used as features to subsequent classifiers for the subsequent target labels.
- **Label Powerset**: This is an extreme case of problem transformation wherein all possible combinations of target labels are taken into consideration. These labels are considered as unique labels for a multiclass classification problem. Thus, one classifier is trained to predict one of these unique labels.

[3] https://huggingface.co/bert-base-uncased.

[4] https://yknzhu.wixsite.com/mbweb.

[5] https://en.wikipedia.org/wiki/English_Wikipedia.

Table 6
Description of the evaluation metrics used.

Metric	Description	Formula
Hamming loss (HL)	It is the fraction of incorrectly classified labels	$HL = \frac{1}{\lvert N\rvert \cdot \lvert L\rvert} \sum_{i=1}^{\lvert N\rvert} \sum_{j=1}^{\lvert L\rvert} y_{i,j} \oplus z_{i,j}$
Jaccard similarity coefficient (J)	It is also known as intersection over Union and gives us a measure of similarity between predicted labels and ground truth	$J(A, B) = \frac{\lvert A\cap B\rvert}{\lvert A\cup B\rvert} = \frac{\lvert A\cap B\rvert}{\lvert A\rvert+\lvert B\rvert-\lvert A\cap B\rvert}$, where $0 <= J(A, B) <= 1$
0/1 Loss (L)	This gives us the fraction of misclassifications	$L(\hat{y}, y) = I(\hat{y} \neq y)$, where I is the indicator function
Mean average precision (MAP)	It is the mean of the average precision scores for each query	$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$ where Q is the number of queries
Macro F_1 score	It is the harmonic mean of Macro Precision and Macro recall of the n classes	$MaP = \frac{\sum_{i=1}^{n} Precision_i}{n}$, $MaR = \frac{\sum_{i=1}^{n} Recall_i}{n}$, $MacroF_1Score = 2.\frac{MaP*MaR}{MaP+MaR}$
Micro F_1 score	It is the harmonic mean of Micro Precision and Micro recall of the n classes	$MiP = \frac{\sum_{i=1}^{n} Precision_i}{n}$, $MiR = \frac{\sum_{i=1}^{n} Recall_i}{n}$, $MicroF_1Score = 2.\frac{MiP*MiR}{MiP+MiR}$

5.2. InPHYNet model hyperparameter settings

InPHYNet is trained to perform multi-label classification for the problem setting. The primary task is the multi-label prediction for physics K-12 text for which the primary dataset "Multi-label Physics K-12 dataset" (described in Section 3.1) is used. Since only one auxiliary task is used, n denoting the number of auxiliary tasks is set to one. The auxiliary task is the multi-class prediction of question types for which the "Experimental Data for Question Classification" (described in Section 3.2) is used.

Experiments are performed by keeping all model hyperparameters constant except those of the preprocessing layer. The input embedding layers of all the tasks (primary and auxiliary) use a fixed length input encoding vector of 500 dimensions. The output vectors from the embedding layers are passed into the primary and auxiliary unidirectional LSTMs. The primary unidirectional LSTM uses 512 dimensional hidden and cell states. For the auxiliary unidirectional LSTM, 512 dimensional hidden and cell states are used. The auxiliary transfer unidirectional LSTM uses 512 dimensional hidden and cell states similar to that of the auxiliary unidirectional LSTMs. The *weight alignment layer* is a two-layer fully connected feed-forward neural network with 1024 neurons in each layer. Sigmoid activation function is used for both first and second layer of the neurons.

A softmax layer is used as the output classification layer for the auxiliary task. The *information aggregation layer* is a two-layer network. The first layer in the *information aggregation layer* is a fully-connected linear layer that outputs a fixed length vector corresponding to the output dimension. This vector is converted into a probability vector by passing it through a softmax layer. This softmax output is thresholded to obtain the final multi-label outputs. In experiments, two separate pre-processing layers, namely Doc2Vec and TF–IDF vectors are used. Since our textual data domain is primarily school-level Physics, we did not want our model to be biased due to any out-of-domain vocabulary sets either through the use of auxiliary task datasets or pre-trained language models such as BERT, ELMO and MT-DNN. We therefore used only the non pre-trained TF–IDF and Doc2Vec models. Both the TF–IDF and Doc2Vec pre-processing layers are trained and validated solely on our primary task physics dataset. The vocabulary size for the TF–IDF vectors for the primary Physics dataset is 9743. Analysis is done on the results obtained by two pre-processing layers separately.

For each of the two separate pre-processing layer configurations, the InPHYNet model is trained for 50 epochs. Adam optimization strategy is used to help achieve faster convergence. A batch size of 64 is used to process the input data samples. Cross-entropy loss function is used to perform the backpropagation. The loss function plots are shown in Fig. 7. Gradient clipping is applied to avoid exploding gradients which could affect the optimization of the model. All the gradients are clipped at a maximum value of 10. A learning rate of 0.001 is used. The final loss is computed with a μ_1 value of 0.5 for the auxiliary task (Eq. (16)):

$$NetLoss = PrimaryLoss + 0.5 \times AuxLoss. \tag{17}$$

5.3. Evaluation metrics

As described by Wu and Zhou [65], the models are evaluated using six different metrics that are most prevalent for multi-label classification. The evaluation metrics used are listed in Table 6.

6. Experimental results

Results of the multi-label classification are shown for each of the two preprocessing layers used (as explained in Section 4.1). For each preprocessing layer setting, results are reported for eleven best baseline machine learning models and three deep learning models: Vanilla LSTM classifier, BERT and the GIRNet model along with the proposed InPHYNet model. Results of doc2vec are shown in Table 7 and of TF–IDF vector are shown in Table 8.

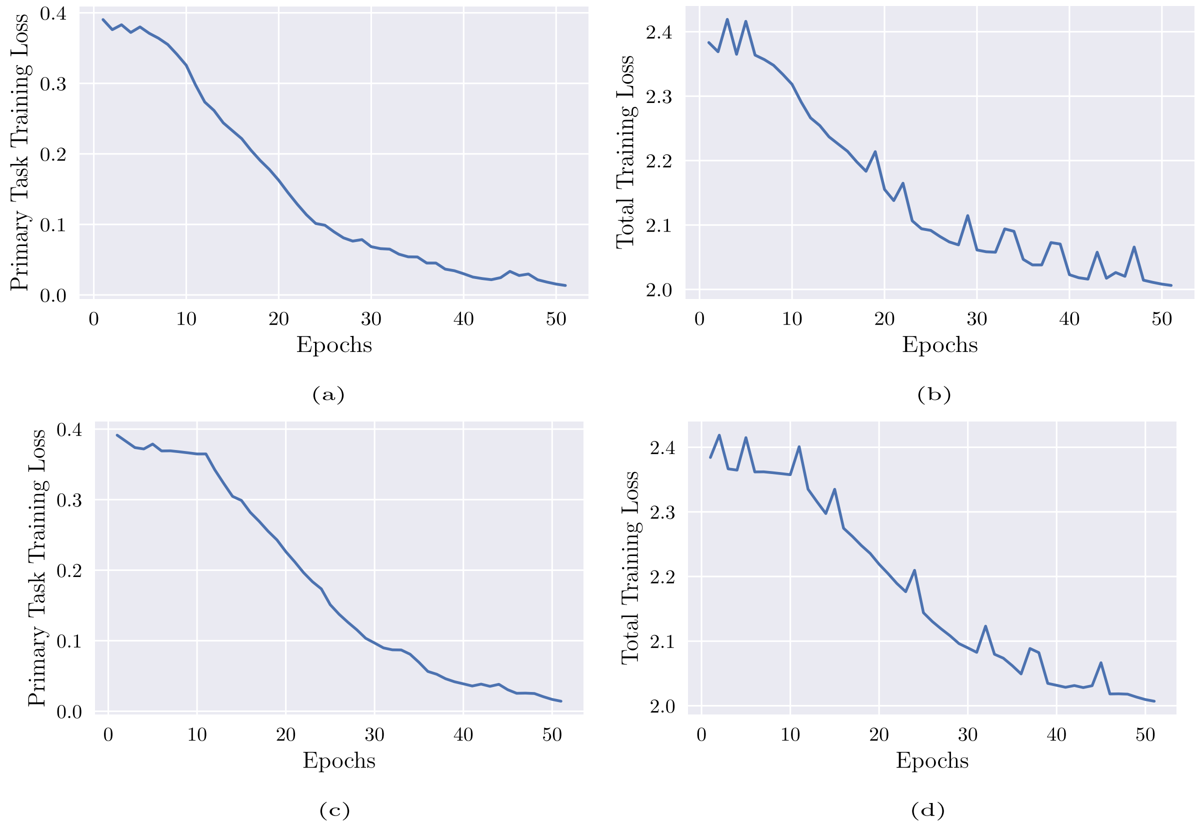
We observe that InPHYNet with the TF–IDF pre-processing outperforms the BERT model. However, InPHYNet with the Doc2Vec pre-processing model performs worse than the BERT model. We believe that this is because of the Doc2Vec's inability to learn high quality paragraph representations for unseen data. Further, as portrayed by the TF–IDF pre-processing results, the TF–IDF latent representations learnt are more structured and expressive than the Doc2Vec and BERT representations.

As is evident from the results, InPHYNet outperforms all the baseline models and provides best results for all TF–IDF preprocessing methods used.

6.1. Class-wise representations

6.1.1. Relation between grades and concepts difficulty

We performed an analysis on the difficulties of concepts taught to students as they move to senior grade classes. Results of four types of experiments are presented in Table 9. The model is



(a)

(b)

(c)

(d)

Fig. 7. (a)–(b): Plots of the primary and total loss values respectively for model trained with **Doc2Vec** preprocessing. (c)–(d): Plots of the primary and total loss values respectively for model trained with **TF–IDF** preprocessing.

Table 7
Results for doc2vec vector preprocessing of paragraphs. The abbreviations represent problem transformation techniques. BR — Binary Relevance, CC — Classifier Chain, LP — Label Powerset. The values in bold are the best results obtained across all models.

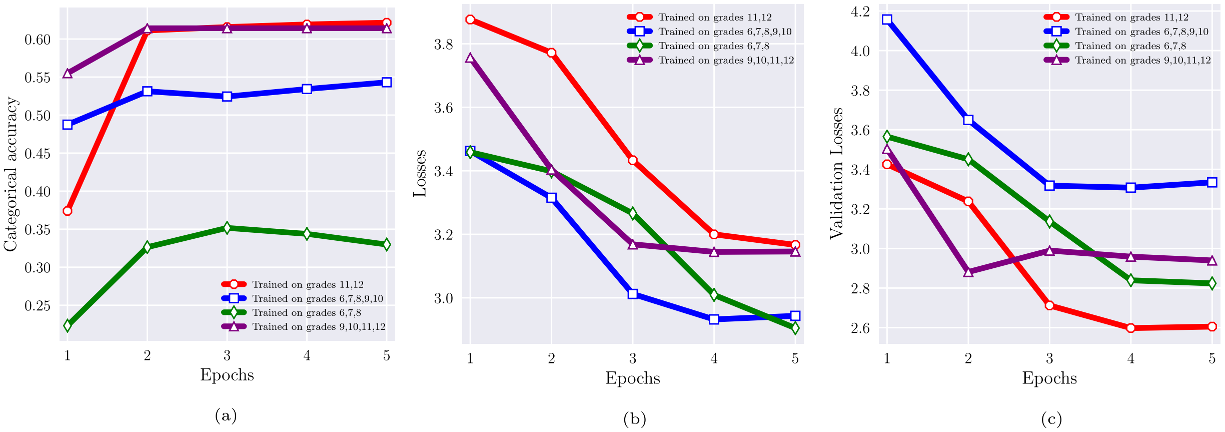
Classifier	Hamming loss	Jaccard similarity score	0/1 loss	Average precision	Macro F_1 score	Micro F_1 score
MLP (sigmoid, 1, 100) - BR	0.17	0.399	0.764	0.225	0.216	0.484
MLP (relu, 1, 100) - BR	0.166	0.354	0.805	0.238	0.252	0.476
MLP (sigmoid, 1, 100) - CC	0.173	0.41	0.757	0.222	0.223	0.487
MLP (relu, 1, 100) - CC	0.18	0.393	0.77	0.227	0.258	0.473
MLP (sigmoid, 1, 100)-LP	0.172	**0.417**	0.746	0.219	0.206	0.488
MLP (relu, 1, 100) - LP	0.175	0.42	0.746	0.226	0.245	0.486
GaussianNB - BR	0.229	0.122	0.981	0.214	0.226	0.338
Decision Tree - CC	0.24	0.294	0.893	0.207	0.25	0.397
Decision Tree - LP	0.23	0.295	0.877	0.204	0.236	0.382
Random Forest - BR	0.169	0.359	0.794	0.216	0.176	0.454
MLkNN20	0.173	0.33	0.825	0.227	0.219	0.449
Vanilla LSTM (512 hidden states)	0.19	0.214	0.765	0.183	0.134	0.353
BERT	**0.119**	**0.42**	**0.656**	0.309	0.312	**0.591**
GIRNet	0.161	0.299	0.752	0.251	0.258	0.46
Proposed InPHYNet	0.135	0.381	0.703	**0.323**	**0.369**	0.552

trained on the material of the grades specified in the first column and is tested on the grades mentioned in the second column. We observe that the model's accuracy is maximum when it is trained on 11th and 12th grades. This is intuitively analogous to how learning happens in school, wherein a student of higher grade would be covering the topics in much more detail and hence, would be able to answer the questions based on concepts of lower grades very well. In contrast to this, when we train our model on 6th to 10th grades and test on 11th and 12th grades, we notice a drop in the accuracy. This is again analogous to how a student learns in school, wherein there are various new and deeper concepts covered in higher grades which would be difficult for the students in lower grades to understand. Thus, our trained models are performing in consonance with how a student learns concepts in school on the intended task of text classification in physics.

The plots for the validation categorical accuracy and loss are shown in Fig. 8. Studies similar to this current work may prove, in particular, very helpful to researchers in the educational field [66] and may help in devising better strategies on curriculum design and pedagogy.

6.1.2. Effects of gradewise training

To assess the representational capacity of conceptual material in each grade, we created seven separate training sets $T_6, T_7, T_8, T_9, T_{10}, T_{11}, T_{12}$, where T_i represents the ith grade's training data. The number of training samples in each training set are kept equal to avoid any noise or bias within the training data across grades. A separate set of fixed size was held out.

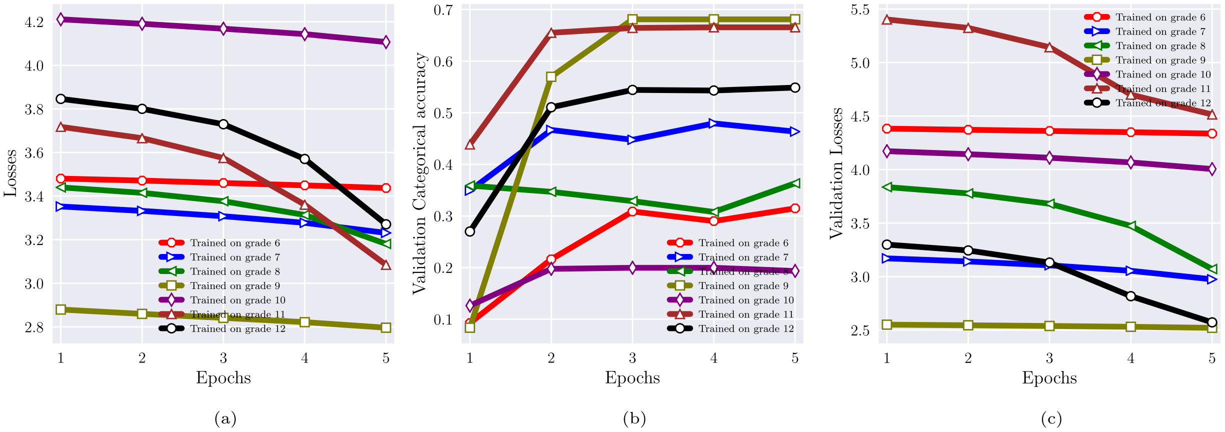


(a)

(b)

(c)

Fig. 8. (a) Plot of training loss vs. epochs, (b) Plot of validation categorical accuracy vs. epochs, (c) Plot of validation loss vs. epochs.



(a)

(b)

(c)

Fig. 9. (a) Plot of validation categorical accuracy vs. epochs, (b) Plot of training loss vs. epochs, (c) Plot of validation loss vs. epochs.

Table 8
Results for TF–IDF vector preprocessing of paragraphs. The abbreviations represent problem transformation techniques. BR — Binary Relevance, CC — Classifier Chain, LP — Label Powerset. The values in bold are the best results obtained across all models.

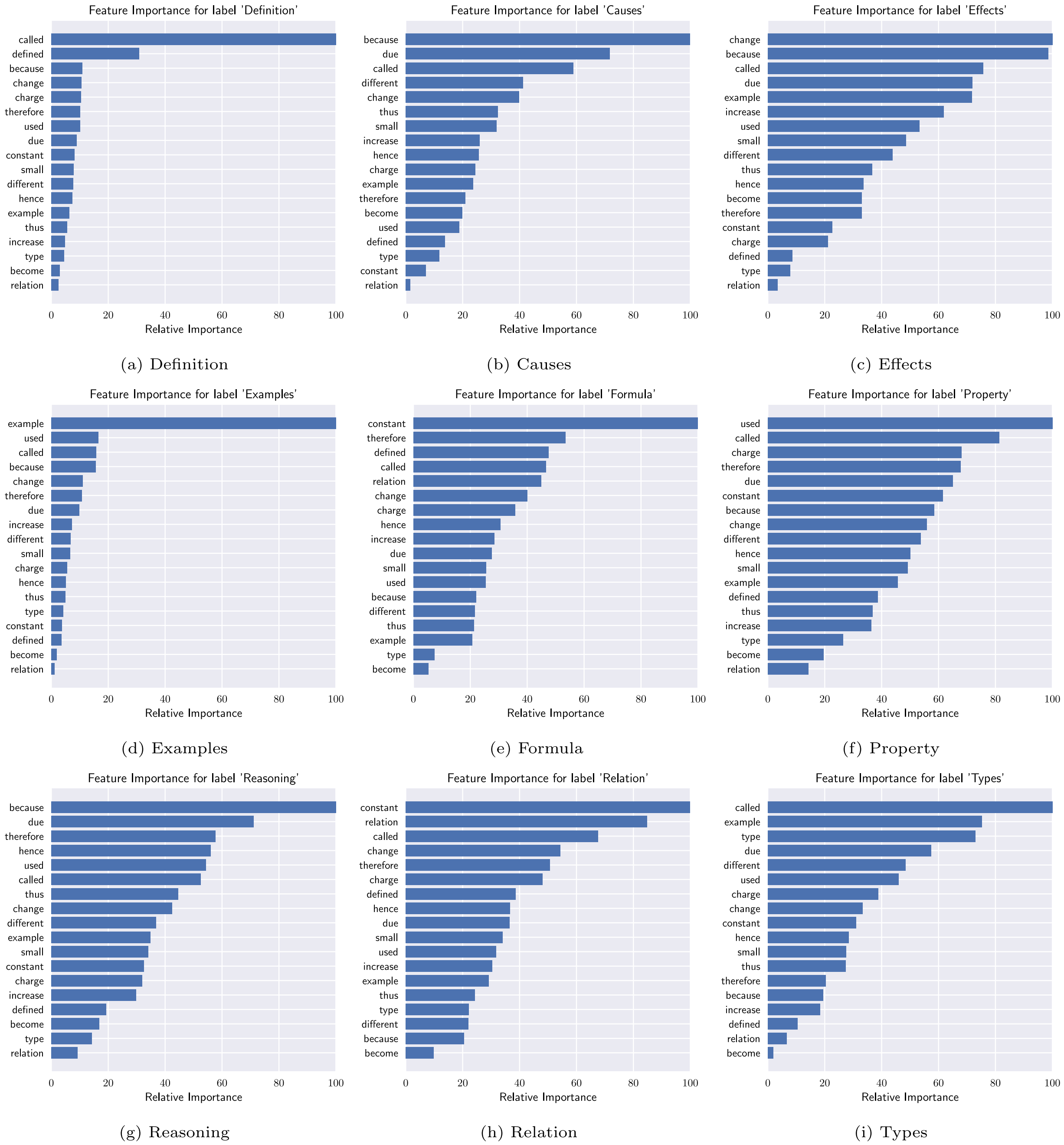
Classifier	Hamming loss	Jaccard similarity score	0/1 loss	Average precision	Macro F_1 score	Micro F_1 score
MLP (sigmoid, 1, 100) - BR	0.146	0.406	0.767	0.246	0.269	0.519
MLP (relu, 1, 100) - BR	0.132	0.483	0.695	0.299	0.386	0.592
MLP (sigmoid, 1, 100) - CC	0.15	0.462	0.719	0.245	0.284	0.539
MLP (relu, 1, 100) - CC	0.144	**0.508**	0.675	0.282	0.382	0.57
MLP (sigmoid, 1, 100) - LP	0.154	0.462	0.705	0.229	0.25	0.517
MLP (relu, 1, 100) - LP	0.151	0.484	0.695	0.26	0.336	0.55
GaussianNB - BR	0.61	0.252	0.984	0.2	0.306	0.365
Decision Tree - CC	0.197	0.377	0.821	0.229	0.323	0.468
Decision Tree - LP	0.195	0.399	0.79	0.229	0.319	0.476
Random Forest - BR	0.133	0.462	0.699	0.266	0.29	0.557
MLkNN20	0.145	0.438	0.737	0.252	0.306	0.54
Vanilla LSTM (512 hidden states)	0.191	0.236	0.734	0.26	0.237	0.382
BERT	0.119	0.42	0.656	0.309	0.312	0.591
GIRNet	0.127	0.429	0.617	0.341	**0.451**	0.6
Proposed InPHYNet	**0.109**	**0.462**	**0.547**	**0.381**	0.424	**0.632**

This set contains equal number of data samples from each of the seven grades' training data. In this experiment, the held out test set contains 350 data samples (50 samples from each of the seven classes). We trained InPHYNet independently on each of the grade-wise training sets T_i. We analyzed the obtained categorical losses and validation set accuracy to determine the quality of each of the seven trained models. We also compared the accuracy obtained by each of the grade-wise models on the held-out test set to understand the representational power of the data present in every grade's training set (Table 10).

Table 9
Results depicting the relationship between grades and concept difficulty.

Grades trained on	Grades tested on	Validation accuracy	Test accuracy
11 , 12	6, 7, 8, 9, 10	0.64	0.61
6, 7, 8	9, 10, 11, 12	0.31	0.28
6, 7, 8, 9, 10	11, 12	0.54	0.50
9, 10, 11, 12	6, 7, 8	0.61	0.57

We observe that the models trained on 11th and 12th grades achieve very high accuracy on the test set. This can be intuitively attributed to the fact that the training sets of 11th and



(a) Definition

(b) Causes

(c) Effects

(d) Examples

(e) Formula

(f) Property

(g) Reasoning

(h) Relation

(i) Types

Fig. 10. Plots depicting the feature importance of certain specific vocabulary terms for each label type.

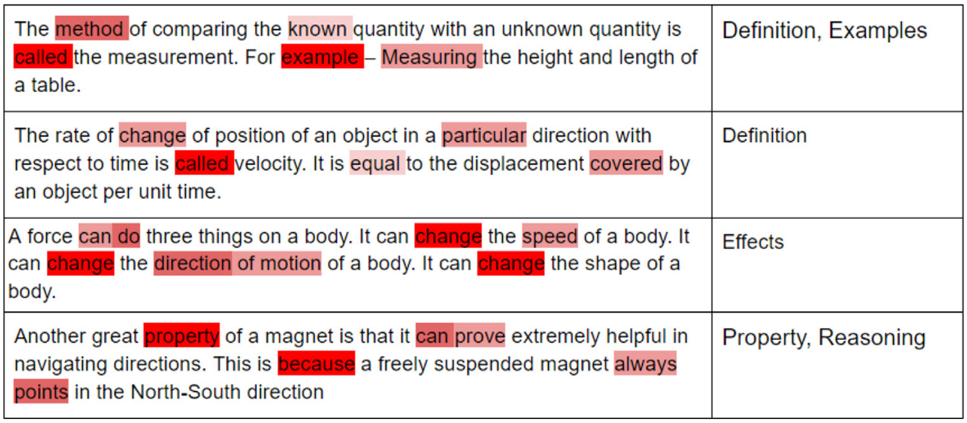
Table 10
Results depicting the effect of gradewise training experiment.

Grades trained	Test accuracy
6	0.31
7	0.44
8	0.39
9	0.67
10	0.2
11	0.65
12	0.55

12th grades have the higher power of concept representation compared to other grades. This is because 11th and 12th grades have the highest requirement of conceptual understanding. This is analogous to the scenario wherein a 11th grade or a 12th grade student is much better equipped to understand concepts rather than a student from a lower grade class.

Another interesting observation is that the model trained on 9th grade achieved highest accuracy. We attribute this property to the wide variety of chapters present in the training data of grade 9. Due to varied chapter representation, the model is able to understand a wide range of contextual information across all grades and hence, is able to perform very well on the test set.

One particular anomaly that we notice is the surprisingly low accuracy of the model trained on grade 10. We hypothesize that this may either be due to an unfortunate split of the training data of grade 10 or the level of constriction of chapters in the grade 10 material. The plots for the validation categorical accuracies and losses are shown in Fig. 9.



The method of comparing the known quantity with an unknown quantity is called the measurement. For example – Measuring the height and length of a table.	Definition, Examples
The rate of change of position of an object in a particular direction with respect to time is called velocity. It is equal to the displacement covered by an object per unit time.	Definition
A force can do three things on a body. It can change the speed of a body. It can change the direction of motion of a body. It can change the shape of a body.	Effects
Another great property of a magnet is that it can prove extremely helpful in navigating directions. This is because a freely suspended magnet always points in the North-South direction	Property, Reasoning

Fig. 11. Heat map depicting the importance of words in four different input paragraphs and their corresponding labels. The darker the shade of red of a word, the more important it is for the label prediction. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6.2. Feature importance

In order to better understand the model and to gain an intuition of the importance of each vocabulary term for a particular label, we plot the relative feature importance of a few commonly occurring terms (refer to Fig. 10). To do this, we consider only the feature vectors generated by these vocabulary terms as data and the corresponding annotations of a particular label as the labels and compute their feature importance. This plot depicts the importance of the chosen terms for determining a particular label. We see that the word most related to the label gets the maximum importance in almost every case. For instance, the words “called” and “defined” have the maximum importance for the label *Definition*. Likewise, word “example” has the maximum importance for the label *Examples*.

6.3. Heatmap

We also generated a heat map depicting the significance of terms within an example paragraph. This gave us a clear understanding of the importance of each word in a paragraph and in determining the set of labels for that particular paragraph. To visualize this, we studied the alignment vector weights learned for the corresponding input paragraph. A higher value represents more significance i.e. more importance of a particular term for the primary task. Some examples of heatmap representations are shown in Fig. 11. The importance is indicated by various shades of red, with darker shades denoting the more important terms and lighter shades representing less important terms.

InPHYNet yields significant accuracy on the dataset and can be used for any education-based text classification/annotation purposes. Thus, InPHYNet can be used as a module within the educational questioning answering systems to improve their performance, similar to the ones proposed by Atapattu et al. [67] and Alzetta et al. [68].

7. Discussion

A vital component of every questioning answering system is a question type classification module. The contributions of our work in this are two fold. Firstly, we have curated a well annotated dataset to carry out this task. Secondly, we propose InPHYNet, an architecture to carry out the multi-label text classification task, setting up a new standard for question type classification specific in the educational domain.

7.1. Implication of this work

This work can have far reaching impact in the area of education. As noted in Section 2.1, a lot of work has been done in the area of education for enhancing the learning experience of students using technology. A lot of existing work in the domains of intelligent tutoring systems, question answering systems and response evaluation modules can be leveraged along with the proposed InPHYNet architecture to build high performance educational question-answering systems and tutoring agents. Such systems can be deployed in educational institutions to reduce the burden on teachers and improve the learning experiences of students. This will help the educational systems as a whole, generally, overwhelmed with the increasing number of students, while enhancing the learning of students.

7.2. Limitations

The current work suffers with some limitations summarized as below.

- **Limited size of our self curated dataset:** In order to make the most of deep learning models, a primary need is a large data set. But there is a dearth of well annotated data with respect to text classification in education. For example, in the absence of a curated dataset for physics education, we prepared our inhouse dataset. Although we used multitask learning network to obtain reliable results with our small inhouse dataset, there is a need of the availability of large curated dataset for different school subjects for building intelligent tutoring systems. This can really boost research worldwide in this area.
- **Focus on text only:** Our dataset focuses on the textual part of the documents. However, diagrams and figures form an important aspect of teaching subjects such as physics. There is a need of a dataset that also includes images and videos.
- **Dataset built for physics:** Our study focused on grade 6–12 physics. However, with evident lack of well-curated dataset of educational texts, it may be preferable to expand this dataset to include more subjects.
- **Focus on CBSE curriculum:** Our dataset and study focused on the CBSE curriculum for the data. But, worldwide and even within India, different curriculum are followed. Thus, there is a need to widen this dataset for even school physics.
- **Using InPHYNet in other tasks:** The proposed architecture, InPHYNet can be used for other text classification tasks with focus on educational domain.

- **Construction of an end-to-end QA system:** Taking the motivation for this work forward, we can build a QA system wherein this classification task helps us in two ways: *(i)* It can help reduce the human dependency on manual annotation of the dataset; and *(ii)* It can form an essential part of the document retrieval unit by helping develop an understanding of the context of the text in the document.

7.3. Future work

In the future, we would like to explore the use of better latent representations for paragraphs. This could lead to better representational capacity of the network architecture. We would also like to build a network based solely on self attention rather than the traditional LSTM blocks in our network. We believe this would help us in learning shared features between the primary and auxiliary tasks at a larger scale and would require lesser amount of network training time. One potential architecture could be the Transformer Network proposed by Vaswani et al. [69]. Further, in the near future, this work can potentially be used for enhancing the quality of question-answering systems in the educational domain.

8. Conclusion

In this paper, we present a new architecture namely, InPHYNet, a generic question type classification module for the multilabel classification of paragraphs of school physics of 6th to 12 grade. The proposed work can be utilized to build automated intelligent tutoring or question-answering systems for education domain. The physics multi-label classification dataset can also be used as an auxiliary task in conjunction with the other primary text classification task to improve the accuracy of the primary task. This can particularly be exploited in educational domain question and answer type classification problem settings.

CRediT authorship contribution statement

Vishaal Udandarao: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data collection, Data annotation, Data curation, Writing - Original draft. **Abhishek Agarwal:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data collection, Data annotation, Data curation, Writing - Original draft. **Anubha Gupta:** Conceptualization, Methodology, Resources, Writing - review & editing, Supervision, Project administration. **Tanmoy Chakraborty:** Methodology, Software, Formal analysis, Writing - review, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Mohit Rathore for help in validating and testing the models. The authors would also like to thank Vrinda Mittal (Department of CSAM, IIIT Delhi) for helping in the initial exploratory data analysis and collecting the dataset. We would also like to thank the Infosys Centre for AI for partial support for this research work. T. Chakraborty would like to thank the support of the Ramanujan Fellowship (SERB, DST, India).

References

[1] R. Noonan, STEM Jobs: 2017 Update. ESA Issue Brief# 02-17, US Department of Commerce, 2017.

[2] E. Seymour, Talking About Leaving: Why Undergraduates Leave the Sciences, Westview Press, 1997.

[3] M. Biggers, A. Brauer, T. Yilmaz, Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers, ACM SIGCSE Bull. 40 (1) (2008) 402–406.

[4] L. Nadelson, C.M. Sias, A. Seifert, Challenges for integrating engineering into the k-12 curriculum: Indicators of k-12 teachers' propensity to adopt innovation, in: 2016 ASEE Annual Conference & Exposition, 2016.

[5] T.R. Kelley, J. Knowles, A conceptual framework for integrated STEM education, Int. J. STEM Educ. 3 (1) (2016) 11.

[6] T.J. Moore, M.S. Stohlmann, H.H. Wang, K.M. Tank, A.W. Glancy, G.H. Roehrig, Implementation and integration of engineering in k-12 STEM education, in: Engineering in Pre-College Settings: Synthesizing Research, Policy, and Practices, Purdue University Press, 2014, pp. 35–60.

[7] C. Wladis, A.C. Hachey, K. Conway, An investigation of course-level factors as predictors of online stem course outcomes, Comput. Educ. 77 (2014) 145–150.

[8] M. Dzikovska, N. Steinhauser, E. Farrow, J. Moore, G. Campbell, Beetle ii: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics, Int. J. Artif. Intell. Educ. 24 (3) (2014) 284–332.

[9] I.A. Halloun, D. Hestenes, The initial knowledge state of college physics students, Amer. J. Phys. 53 (11) (1985) 1043–1055.

[10] R.R. Hake, Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses, Amer. J. Phys. 66 (1) (1998) 64–74.

[11] R.E. Mayer, Learning and Instruction, Prentice Hall, 2003.

[12] C. Wieman, K. Perkins, Transforming physics education, Phys. Today 58 (11) (2005) 36.

[13] F. Noorbehbahani, A.A. Kardan, The automatic assessment of free text answers using a modified bleu algorithm, Comput. Educ. 56 (2) (2011) 337–345.

[14] W. Westera, M. Dascalu, H. Kurvers, S. Ruseti, S. Trausan-Matu, Automated essay scoring in applied games: Reducing the teacher bandwidth problem in online training, Comput. Educ. 123 (2018) 212–224.

[15] F. Rodrigues, P. Oliveira, A system for formative assessment and monitoring of students' progress, Comput. Educ. 76 (2014) 30–41.

[16] T. Atapattu, K. Falkner, N. Falkner, A comprehensive text analysis of lecture slides to generate concept maps, Comput. Educ. 115 (2017) 96–113.

[17] A. Agarwal, N. Sachdeva, R.K. Yadav, V. Udandarao, V. Mittal, A. Gupta, A. Mathur, EDUQA: Educational domain question answering system using conceptual network mapping, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 8137–8141.

[18] S. Afzal, T. Dhamecha, N. Mukhi, R. Sindhgatta, S. Marvaniya, M. Ventura, J. Yarbro, Development and deployment of a large-scale dialog-based intelligent tutoring system, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 2, Association for Computational Linguistics, 2019, pp. 114–121.

[19] T.I. Dhamecha, S. Marvaniya, S. Saha, R. Sindhgatta, B. Sengupta, Balancing human efforts and performance of student response analyzer in dialog-based tutors, in: International Conference on Artificial Intelligence in Education, Springer, 2018, pp. 70–85.

[20] S. Marvaniya, S. Saha, T.I. Dhamecha, P. Foltz, R. Sindhgatta, B. Sengupta, Creating scoring rubric from representative student answers for improved short answer grading, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 993–1002.

[21] R. Caruana, Multitask learning, Mach. Learn. 28 (1) (1997) 41–75.

[22] H.G. Noushahr, S. Ahmadi, Multitask learning for text classification with deep neural networks, in: International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, 2016, pp. 119–133.

[23] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, 2016, arXiv preprint arXiv:1607.01759.

[24] A. Benton, M. Mitchell, D. Hovy, Multi-task learning for mental health using social media text, 2017, arXiv preprint arXiv:1712.03538.

[25] P. Liu, X. Qiu, X. Huang, Recurrent neural network for text classification with multi-task learning, 2016, arXiv preprint arXiv:1605.05101.

[26] M. Rei, Semi-supervised multitask learning for sequence labeling, 2017, arXiv preprint arXiv:1704.07156.

[27] Y. Zhang, Q. Yang, A survey on multi-task learning, 2017, arXiv preprint arXiv:1707.08114.

[28] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473.

[29] M. Lan, J. Wang, Y. Wu, Z.-Y. Niu, H. Wang, Multi-task attention-based neural networks for implicit discourse relationship representation and identification, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1299–1308.

[30] S. Liu, E. Johns, A.J. Davison, End-to-end multi-task learning with attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1871–1880.

[31] A.C. Stickland, I. Murray, Bert and pals: Projected attention layers for efficient adaptation in multi-task learning, 2019, arXiv preprint arXiv:1902.02671.

[32] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[33] Y. Zhang, P. Zhang, Y. Yan, Attention-based LSTM with multi-task learning for distant speech recognition., in: Interspeech, 2017, pp. 3857–3861.

[34] D. Gupta, T. Chakraborty, S. Chakrabarti, Girnet: Interleaved multi-task recurrent state sequence models, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 6497–6504.

[35] L.S. Larkey, W. Croft, Combining classifiers in text categorization, in: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996, pp. 289–297.

[36] D.D. Lewis, W.A. Gale, A sequential algorithm for training text classifiers, in: SIGIR'94, Springer, 1994, pp. 3–12.

[37] T.M. Mitchell, Evaluating hypotheses, Mach. Learn. (1997) 128–153.

[38] A. McCallum, K. Nigam, et al., A comparison of event models for naive bayes text classification, in: AAAI-98 Workshop on Learning for Text Categorization, Vol. 752, Citeseer, 1998, pp. 41–48.

[39] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, J. Mach. Learn. Res. 2 (Nov) (2001) 45–66.

[40] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, M. Demirbas, Short text classification in twitter to improve information filtering, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2010, pp. 841–842.

[41] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, 2015, pp. 649–657.

[42] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[43] C. Zhou, C. Sun, Z. Liu, F. Lau, A c-lstm neural network for text classification, 2015, arXiv preprint arXiv:1511.08630.

[44] H. Tao, S. Tong, H. Zhao, T. Xu, B. Jin, Q. Liu, A radical-aware attention-based model for chinese text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 5125–5132.

[45] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.

[46] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, 2018, arXiv preprint arXiv:1802.05365.

[47] X. Liu, P. He, W. Chen, J. Gao, Multi-task deep neural networks for natural language understanding, 2019, arXiv preprint arXiv:1901.11504.

[48] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems, 2019, pp. 5754–5764.

[49] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019, arXiv preprint arXiv:1907.11692.

[50] F. Herrera, F. Charte, A.J. Rivera, M.J. Del Jesus, Multilabel classification, in: Multilabel Classification, Springer, 2016, pp. 17–31.

[51] R.B. Pereira, A. Plastino, B. Zadrozny, L.H. Merschmann, Categorizing feature selection methods for multi-label classification, Artif. Intell. Rev. 49 (1) (2018) 57–78.

[52] N.A. Ahmed, M.A. Shehab, M. Al-Ayyoub, I. Hmeidi, Scalable multi-label arabic text classification, in: 2015 6th International Conference on Information and Communication Systems (ICICS), IEEE, 2015, pp. 212–217.

[53] W.-C. Chang, H.-F. Yu, K. Zhong, Y. Yang, I. Dhillon, A modular deep learning approach for extreme multi-label text classification, 2019, arXiv preprint arXiv:1905.02331.

[54] S. Baker, A.-L. Korhonen, Initializing Neural Networks for Hierarchical Multi-Label Text Classification, Association for Computational Linguistics, 2017.

[55] M. Li, Z. Fei, M. Zeng, F.-X. Wu, Y. Li, Y. Pan, J. Wang, Automated ICD-9 coding via a deep learning approach, IEEE/ACM Trans. Comput. Biol. Bioinform. 16 (4) (2018) 1193–1202.

[56] F. Gargiulo, S. Silvestri, M. Ciampi, Deep convolution neural network for extreme multi-label text classification., in: HEALTHINF, 2018, pp. 641–650.

[57] X. Li, D. Roth, Learning question classifiers, in: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 2002, pp. 1–7.

[58] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.

[59] A. Graves, G. Wayne, I. Danihelka, Neural turing machines, 2014, arXiv preprint arXiv:1410.5401.

[60] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, in: Proceedings of the First Instructional Conference on Machine Learning, Vol. 242, Piscataway, NJ, 2003, pp 133–142.

[61] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents. ICML: 1188–1196, Google Sch. Google Sch. Digit. Libr. Digit. Libr. (2014).

[62] B. Wang, K. Liu, J. Zhao, Inner attention based recurrent neural networks for answer selection, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1288–1297.

[63] M.-L. Zhang, Z.-H. Zhou, Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

[64] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2009, pp. 254–269.

[65] X.-Z. Wu, Z.-H. Zhou, A unified view of multi-label performance measures, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 3780–3788.

[66] D. Szűcs, U. Goswami, Educational neuroscience: Defining a new discipline for the study of mental representations, Mind Brain Educ. 1 (3) (2007) 114–127.

[67] T. Atapattu, K. Falkner, N. Falkner, Educational question answering motivated by question-specific concept maps, in: International Conference on Artificial Intelligence in Education, Springer, 2015, pp. 13–22.

[68] C. Alzetta, G. Adorni, I. Celik, F. Koceva, I. Torre, Toward a user-adapted question/answering educational approach, in: Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, 2018, pp. 173–177.

[69] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.