Integrating deep learning to improve text understanding in conversation-based ITS

Sheng Xu

Central China Normal University,
Wuhan, China
Email: psyxusheng@mails.ccnu.edu.cn

Frank Andrasik

Department of Psychology,
University of Memphis,
Memphis, TN 38152, USA
Email: fndrasik@memphis.edu

Zhiqiang Cai

University of Wisconsin – Madison,
Wisconsin, USA
Email: zhiqiang.cai@wisc.edu

Xiangen Hu*

Department of Psychology,
The University of Memphis,
Memphis, TN 38152, USA
Email: xhu@memphis.edu
*Corresponding author

Abstract: In Conversation-based intelligent tutoring systems (CbITS), assessing learners' natural language input is a key factor for the system to be effective. When using AutoTutor, a well-known CbITS, assessments of this type are reduced to evaluating the semantic similarity between learners' inputs and pre-set expectations/misconceptions. Traditional semantic representation methods have prominent inherent limitations, while more advanced deep learning models require large amounts of labelled data which is expensive to obtain. We contend that using deep learning models in concert with an active learning training procedure can reduce the demand for labelled data, thus improving the effectiveness of natural language understanding in CbITS. We report findings from a series of experiments that document how our proposed model was able to significantly outperform traditional models with much fewer labelled data. These findings thus illustrate both the possibility and potential benefits that can be accrued by utilising more advanced semantic representation models.

Keywords: conversation-based ITS; semantic; deep learning; active learning; pretrained language model.

Copyright © 2021 Inderscience Enterprises Ltd.

Reference to this paper should be made as follows: Xu, S., Andrasik, F., Cai, Z. and Hu, X. (2021) 'Integrating deep learning to improve text understanding in conversation-based ITS', *Int. J. Smart Technology and Learning*, Vol. 2, No. 4, pp.304–324.

Biographical notes: Sheng Xu is currently a doctoral candidate at Central China Normal University (CCNU). He was trained in Mathematics and his research interests include applied mathematics, cognitive psychology, and semantic analysis.

Frank Andrasik is a Distinguished Professor & former Chair of Psychology & an Affiliate Faculty Member of the Institute for Intelligent Systems, University of Memphis, TN. He served as a Senior Research Scientist at the Florida Institute for Human & Machine Cognition, Pensacola, FL, among other positions. He is a Fellow in 7 professional societies. In 2018 he was selected by the UofM Board of Visitors to receive the Willard R. Sparks Eminent Faculty Award, the highest distinction given to a faculty member. He has published over 300 articles & chapters & 8 co-edited/co-authored texts (with an h-index of 59).

Zhiqiang Cai is currently a researcher in the Wisconsin Center for Education Research (WCER), University of Wisconsin-Madison. He was a research assistant professor at IIS, the University of Memphis from 2001 to 2019, and an assistant/associate professor at the department of mathematics, Huazhong University of Science and Technology, China from 1985 to 2001. His research interests include applied mathematics, AI, and Intelligent Tutoring Systems.

Xiangen Hu is currently a professor of psychology, an affiliated professor of electrical & computer engineering, and a professor of computer sciences at the University of Memphis (UofM). Has been at UoM since 1993. He is also a guest professor of Central China Normal University (CCNU) since 2012. Currently serving as the Dean of School of Psychology at CCNU. His research interests include mathematical modelling of psychology, AI, semantic representation and analysis, intelligent tutoring systems. He is also the director of the Advanced Distributed Learning (ADL) University of Memphis Partnership Lab.

1 Introduction

Over the past three decades, research in intelligent tutoring systems (ITS) has made great progress, particularly so for those regarded as tutorial dialogue systems (a special kind of ITS that tutors students by imitating the conversational behaviour of human tutors). At present, many excellent ITSs have been developed and applied in a range of domains, with impressive outcomes achieved. Prominent examples include Cognitive Tutor in k12 math (Ritter et al., 2007), CIRCSIM-Tutor in Physiology (Evens et al., 2002), and ITSPOKE in conceptual mechanics (Litman and Silliman, 2004) AutoTutor has been applied in the domains of computer literacy (Graesser et al., 2003, 2004), conceptual physics (Matthews et al., 2010), biology (Graesser et al. 2012), and critical thinking (Halpern et al., 2012; Millis et al., 2011). Rigorous evaluations have shown that these ITSs have greatly enhanced learning outcomes and, in some cases, the learning gains have been found to be comparable to those obtained with a human tutor (Vanlehn, 2011).

Understanding a student's natural language input is a key factor for enabling tutorial dialogue systems to be effective. For example, Rosé and Vanlehn (2005) pointed out that the natural language understanding approaches used thus far in ITSs fall into two categories: (1) *shallow* approaches that capture the surface semantic feature of student's input, for example pattern matching (Glass, 2001) and the "bag-of words" model, such as latent semantic analysis (Dumais, 2013) in Graesser et al. (2004); and (2) *deep* approaches, which capture deep semantic features, that are usually captured by converting natural language into some kind of logic form, for example semantic parsing in Aleven and Popescu (2003) and Dzikovska et al. (2010). Thus, deep approaches can support evaluations that are more fine-grained and involve more complex semantic relationships. For example, if a student says, "A causes B", then an ITS that employs deep approaches is capable of recognising when a student misunderstands the causality relationship between A and B, which cannot be recognised by an ITS that is based on shallow approaches (Kalliopi-Irini and Robertson, 2002).

Despite the above shortcomings, shallow approaches are still valuable in tutorial dialogue systems, chiefly for the following reasons: 1) as shallow approaches are much easier to build and can provide good performance overall, they may serve as a fallback plan when initial attempts at applying deep approaches fail, such as in Rosé and Vanlehn (2005); 2) the learning domain is ill-defined so that it is difficult to build a deep natural language understander; and 3) most languages do not have the parsers, semantic bases, and knowledge bases needed to construct deep semantic understanders, which leaves shallow approaches as the only viable choice.

Latent semantic analysis (LSA) has been one of the most widely used shallow approaches in the field of tutorial dialogue systems. Compared with the pattern matching method for analysing local information (usually a few words in length), LSA is more often used to evaluate sentence and even paragraph-level answers as a whole. An example might involve determining whether the input belongs to a specific expected answer / expected typical error, as in AutoTutor, a well-known tutorial dialogue system (Person et al., 2001). The dialogue mechanism of AutoTutor is expectation- and misconception-tailored (EMT) based (Graesser et al., 2005). In EMT, a long ideal answer to a deep level question is decomposed into a set of expectations, each representing a specific aspect of the ideal answer. Typical misconceptions to the question are also anticipated. During tutoring, answers provided by a student are constantly compared with those expectations/misconceptions, so that at each step of tutoring, feedback and subsequent steps are adaptively determined based on what expectation/misconceptions are covered by a student. LSA represents a text's semantic value (using real-valued vectors) and computes a texts' semantic similarity (with cosine value) in a way that is relatively simple and effective, with no need for labeled data or supervision (or what is commonly termed "unsupervised machine learning"; Barlow, 1989). However, LSA lacks the ability to represent complex and fine because it ignores word order information (Hu et al., 2007).

Rapid developments in the field of natural language processing over the past decade have greatly improved performance in semantic representation, especially so for the progress of deep learning. It is, therefore, natural to consider replacing LSA with more advanced semantic representation models in order to further improvements. Although such efforts are worthy, they are quite costly, as most of the advanced models are supervised, which requires large amounts of labelled data (as previously discussed). Further, when a specific CbITS application is applied in an ill-defined domain where large, labelled data are not always available, this type of approach is simply not feasible.

Training complex models while using the least amount of labelled data is a frontier challenge in the machine learning/deep learning community, with various methods being explored. Active learning (AL) (Settles, 2009, details later) is one promising method that is increasingly attracting attention. Studies have confirmed that the AL method is beneficial for reducing the dependence on labelled data. However, datasets used in studies conducted to date often either 1) contain only a few semantic categories, such as positive/negative (for example, the movie reviews dataset by Pang and Lee, 2004); or 2) have been constructed for a general purpose in which the difference between semantic categories is relatively large (such as *20 Newsgroups*, n.d.). In a learning context, an ideal answer to a deep level question may contain more aspects than simply positive, negative, or neutral. Further, as learning is highly domain-specific, those aspects may be much harder to distinguish. Theoretically speaking, two main problems when using deep learning models instead of current semantic model (LSA-based) used in AutoTutor can be solved by using an active learning training procedure, which addresses the problem that insufficient labelled training data lead to poor performance of models). Additionally, a deep learning model itself could provide better semantic representation of texts to distinguish domain specific sentences. In this paper, we describe our approach to addressing the limitations noted earlier. Specifically, we review one that combines a deep learning model with active learning training in order to improve text understanding performance. Our goal is to improve the performance of text assessment, while at the same time using the least amount of labelled data necessary. The most appropriate application of our method is to enhance natural language evaluation in CbITS for ill-defined domains. To our knowledge, this is the first time this array of techniques has been applied to CbITS.

2 Related works

2.1 Text matching models

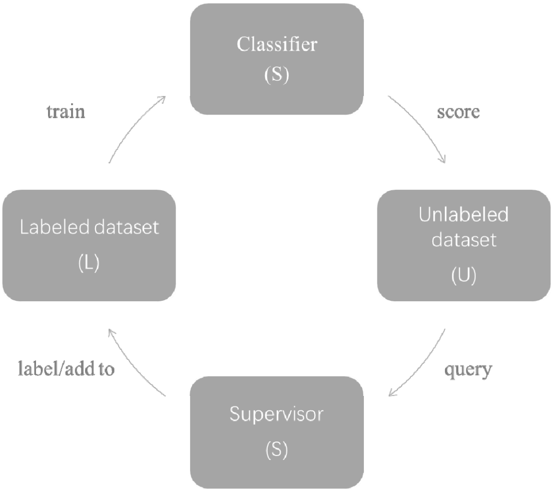
When using an EMT framework, understanding a student's natural language input in a CbITS basically involves determining which input text is best characterised as a pre-set expectation or misconception. This is a typical text classification problem, which involves assigning labels or categories to text according to its content. The key to effective text classification is to better represent the semantics of the text. Several promising approaches have been proposed to better represent a text's semantic meaning, which in general fall into one of three methods. The first is known as the *word embedding* method, with examples being Word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and LSA, among others. These types of models seek to obtain a high-quality vector representation of words. In order to obtain a text's vector, an extra process is needed, such as determining the average or weighted average of word vectors that the text contains (Arora et al., 2016; Rücklé et al., 2018). Word embedding methods are relatively simple and quick. However, in this approach, word order information and the interaction between words is ignored, which limits their ability to capture more precise semantics. The second method is *sentence embedding*, which directly models a sentence's vector by utilising a deep learning structure, such as a convolutional neural network (Kim, 2014) or a recurrent neural network (Nowak et al., 2017) to capture local and global word information, respectively.

The third method consists of pretrained language models (Devlin et al., 2018; Peters et al., 2018; Radford et al., 2018), which are trained on a large-scale corpus using self-supervised tasks so that once generic semantic information is acquired it is capable of being transferred to specific domains by fine-tuning in downstream tasks, which needs minimal data. BERT (Devlin et al., 2018), one of the most highly cited pretrained language models, utilises stacked transformer layers (Vaswani et al., 2017) as its basic neural structure and trains the model on a large-scale corpus that includes both word-level (predicting masked words in a sentence) and sentence-level (determining if two sentences are in succession in an original article) unsupervised tasks. Pretrained language models have achieved state-of-the-art performance on multiple typical NLP tasks, such as that shown by Chen et al. (2019) and others (Rietzler et al., 2019; Yang et al., 2019), which accounts for the high level of attention BERT has enjoyed since its release.

2.2 Active learning

AL consists of a machine learning framework that is used to overcome the labelling bottleneck by asking queries to be labelled by a human annotator. The model for AL is $A = (C, Q, S, L, U)$ (Settles, 2009), where C refers to a classifier, L refers to labelled data, U represents unlabeled data, S refers to a supervisor (where typically, the supervisor is a human annotator with expertise for a given domain), with Q designating the query strategy used for selecting appropriate samples from U. AL is a cyclical and iterative process. In general, C is first trained using L, which is then used to score all samples in U. Based on those scores and query strategy, some samples from U are taken out and labelled by S to be used to train an improved C. This process continues until C's performance is satisfactory or certain stop criteria are met. The entire process of AL is shown in Figure 1. Applying AL has great potential to reduce the need for labelled data, as shown by the work of Miller et al. (2020).

Figure 1 Process of active learning



The most important component for AL is the query strategy, as summarised in Sun and Wang (2010). Uncertainty sampling, in which the active learner queries samples with the least amount of certainty, is the most commonly used query strategy. The basic idea of uncertainty sampling is that the active learner can avoid querying identified samples and focus on confusing instances. Several ways exist to define a sample's uncertainty, such as least confident (Hu et al., 2016) and entropy (Zhang et al., 2016). The entropy-based uncertainty is defined in the following formula:

$$Ent = -\Sigma_k P(prediction = k \mid x;\theta) logP(prediction = k \mid x;\theta)$$

where $P(prediction = k \mid x;\theta)$ means the probability of the model (with parameter *θ*) assigning label *k* to sample *x*. Larger obtained values indicate that the model may not be certain about assigning a specific label to the sentence and is therefore a better candidate to be labelled by human supervisors. This strategy has been used in numerous studies, such as those conducted by Lu and MacNamee (2020), Zhu and Hovy (2007), and Zhu et al. (2008).

Researchers have long tried to use AL to help solve the problem of text classification. For example, consider the work of Tong and Koller (2001), who applied the Support vector machine (SVM). More recently, increased attention has focused on deep learning approaches, such as that by Zhang et al. (2016) who combined AL with a deep learning model (more specifically, the Convolution neural network structure) for text classification. An et al. (2018) compared SVM with deep learning models, such as f(gated recurrent unit, Chung et al., 2014) models, which are variations of recurrent neural networks, and found deep learning models significantly outperform traditional machine learning models. Finally, upon comprehensively comparing many types of text classification models, Lu and MacNamee (2020) found that pretrained language models with uncertainty sampling yielded consistently higher scores.

3 Research questions

A large number of studies have shown the advantages of deep learning models in semantic representation. It is also true that training a deep learning model is resource intensive and expensive (human and computing time), and insufficient training may lead to even worse performance. With this in mind, this paper attempts to address more practical questions, such as:

- How much can the effectiveness be improved by using a pre-training language model + active learning?
- What is the minimum amount of labelled training data needed to train a deep learning text understander that outperforms the model currently used in AutoTutor?

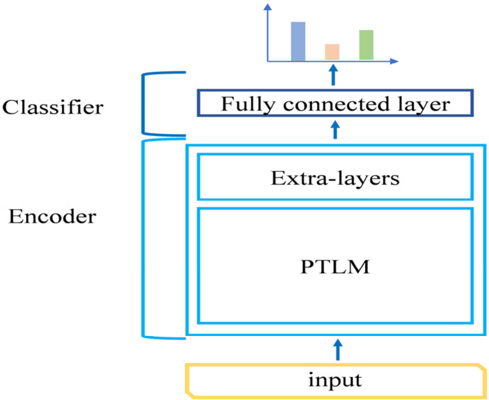
By addressing these questions, this paper seeks to provide practical guidance for the application of deep learning to improve natural language understanding in CbITS.

4 Method

4.1 Model structure

Our model structure is a typical text classification model, which is depicted in Figure 2. It consists mainly of two parts: an encoder that converts text into vectors and a classifier that determines which semantic category best characterises the input. The encoder contains two sub-modules. One consists of a pretrained language model (PTLM) that converts text into an $n \times d$ vector, where n refers to the number of words that the text contains and d refers to the model's output size. The second consists of the extra-layers that operate on the output of PTLM to convert the variable length $n \times d$ vector into a d-dimensional vector. Typical extra layers could consist of a mean pooling layer or a max pooling layer, as detailed in Sun et al. (2019); another classic extra-layer structure includes a bidirectional LSTM with attention (Zhou et al., 2016) or simply just an attention layer alone.

Figure 2 Model proposed. PTLM refers to a pre-trained language model



More specifically, assuming a student's input is sentence s_i containing n words $[w_1, w_2, ..., w_n]$, then through PTLM, it will be converted into an $n \times d$ vector $v_{PTLM} \in R^{n \times d_{PTLM}}$, where d_{PTLM} is the output dimension of PTLM. Mean-pooling and max-pooling both operate on the first dimension of v_{PTLM} by taking the mean value or the maximum value respectively, resulting in a vector $v \in R^{1 \times d_{PTLM}}$. Attention mechanisms in deep learning models can help one to focus on more important information (Bahdanau et al., 2014). We adopt a simple attention mechanism here for the purpose of illustration. Assuming a sentence is converted into a vector $V \in R^{n \times d}$ by all neural layers before the attention layer, then the attention mechanism works as follows:

1 Computing the weighting scores for every word this sentence contains: $s_i = V^{i\cdot} \cdot W_{att} + b_{att}$. W_{att} and b_{att} are learnable parameters, V^i refers to the i-th row in V which can also be regarded as the i-th word's embedding computed by previous layers.

2 Normalising the weighting scores using SoftMax: $a_i = e^{s_i} / \sum_{k=1}^{n} e^{s_k}$;

3 Applying the weighted average to obtain *v*: $v = \sum_{i=1}^{n} a_i . V^{i\cdot}$.

So given a piece of text, the encoder converts it into a 1-dimension vector *v* for representing its semantics. Then the 1-dimension vector is sent to the classifier for classification. For the purpose of illustration, we use a fully connected layer. The final output of the model is $o = Softmax\left(W_o \cdot v + b_o\right) \in R^{1\times N_{cls}}$, where N_{cls} refers to the number of categories in total, while W_o, b_o are the fully connected layer's weights and bias.

4.2 Active-learning training procedure

4.2.1 Training and evaluating

The model's training follows the standard process of AL described in previous sections, including repeatedly: i) training the model using *L*, ii) querying the sample from *U*, and iii) annotating the queried samples and adding them to *L* then returning to step i, until the model's performance is acceptable. In step ii, the model's parameters are updated by a mini-batch gradient descent which involves sampling a mini-batch of data from *L*. In each round of AL, for each semantic category (expectation/misconception), n samples are queried through a sampling strategy from *U*.

4.2.2 Query strategy

We use uncertainty sampling because it is simple and effective. The model's output is a vector $o \in R^{1\times N_{cls}}$. The value on each dimension indicates the possibility that the input belongs to the corresponding semantic category. Because the output is processed by SoftMax, the sum of the values of dimensions is 1, so we consequently use the value of each dimension as the corresponding probability (belongs to corresponding semantic category), and we can then calculate the uncertainty of the output using the previously introduced formula based on entropy as: $u = -\sum_{i}^{N_{cls}} o_i \cdot log o_i$. The samples with the largest *u* values will be queried. In order to ensure the balance of the samples in each category, we choose the k samples with the largest u value for each category.

5 Experiments

5.1 Dataset

In order to be as close to the application scenario of the model as possible, the dataset used in a current study should: contain multiple (more than 3) semantic categories and be domain specific. As most publicly available data sets do not meet these requirements, we built our own, medium-sized dataset. The theme for our created data set is the basic knowledge of diabetes. We constructed the dataset as follows: 1) "Diabetes" was the keyword used to obtain relevant text content (about 4.5Mb in file size) in a well-known online question-and-answer community, which contains about 34k sentences. 2) Sixteen semantic categories related to the topic were sorted by a domain expert as shown in Table 1, with three typical examples given for each semantic category. This information was used as the initial labelled data (for starting the AL) and is considered the "gold standard" dataset in our experiments. 3) We then used a simple semantic similarity program to select the most relevant sentences for each category, manually removing sentences that were not relevant to the domain. This resulted in approximately 2500 sentences. 4) These 2500 sentences were then independently classified by two domain experts, with each sentence placed into one of 16 semantic categories. Sentences judged as not belonging to any identified category were discarded. Before formally classifying these sentences, two domain experts tried to mark 100 sentences selected at random. These experts discussed their inconsistent labels, eventually formed a unified classification standard, and then labelled all of the sentences accordingly. The rater consistency (Cohen's Kappa) coefficient between the two independent domain experts was 0.87 on category level and 0.73 in sub-category level, which is acceptable considering the large number of categories to be classified (16). After formally labelling the dataset, the two domain experts discussed their inconsistent labels and finally came up with a unified result. Finally, 2312 sentences were maintained (the gold standards not included). Eighty percent (80%) of the sentences were used as training data, with the remaining 20% used as our testing dataset. The details of the dataset are shown in Table 1.

Table 1 The dataset used in the following experiments

*Category**	*Sub-category*	*Number*	*Description*
Diabetes	Definition	149	Describes what diabetes is, should include related information
	Description	36	Generally mentioned diabetes as a disease
	Categories	32	Talked about different types of diabetes
	Clinical symptoms	75	Clinical manifestations of diabetes
Epidemiological characteristics		164	
Treatment	Hypoglycaemic drugs	101	Treatment concerning hypoglycaemic drugs
	Insulin related	122	Treatment concerning Insulin drugs
	Diet related	378	
	Exercise related	84	
General	General	238	Generally mentioned about treatment, but could not be categorised to any other sub-categories
Cause of disease	Lifestyle	98	Unhealthy lifestyle causes of diabetes
	Other diseases	78	Caused by diseases
	Heredity	73	
Harm	Harm	276	Its harm to human body
Diagnosis	Diagnosis	172	Clinical diagnosis of diabetes
Related metabolism	Related metabolism	236	Metabolic processes related to blood sugar

Note: *A sample of every category is shown in the appendix.

5.2 Model settings

Having confirmed the advantages of deep learning models over traditional machine learning models, no further discussion of comparisons between them seems needed. Three models were compared: 1) A baseline model, word2vec (for converting sentences into vectors) + KNN (K-nearest neighbours) was used for choosing the best match between sentences and semantic categories, simulating the semantic matching algorithm currently used in AutoTutor; 2) the text classification model trained using whole training data as the upper limit that could be reached; and 3) the text classification model trained in the active learning procedure. A Chinese version of BERT was chosen as our PTLM (contains 11M parameters in total, the output dimension is 768).

For the baseline model, the word2vec model was trained in a simplified Chinese Wikipedia dump, with a vector size of 300 f. Comparisons were also conducted to determine the structure of extra layers, including just mean-pooling, just max-pooling, and just attention layer with bidirectional LSTM + attention layer as described above. These key parameters are shown in Table 2.

Table 2 Key parameters of candidates of extra layers

Model structure	*Extra parameters (parameters in extra-layers)*
BERT + mean pooling	None
BERT + max pooling	None
BERT + attention	Fully connected (input size 768, output size 1)
BERT+Bi-LSTM+ attention	Bi-LSTM (input size 768 output size 256) + attention (fully connected input size 768, output size 1)

5.3 Results

5.3.1 Model performance

The most important question when using deep learning is determining the degree to which the chosen model yields a gain in performance. The data presented in Table 3 indicate that the accuracy of the semantic category judgment was greatly increased by adopting a deep learning model. Compared to the baseline model, the best deep learning model performed significantly better (33% vs. 73%) with the cost of manually labelling about 2500 sentences as training and testing data. Further, the BERT + attention model produced the best results (highest f1 and accuracy score) among all candidates. In the remaining experiments to follow, this structure consequently was adopted as the extra layers of the model's encoder.

5.4 Effectiveness of AL

In order to show the advantages of using the AL training model in a more intuitive manner, we first examined the performance of the model when each semantic category queries 5 samples per round (total query $5 \times 16 = 80$ per round). Table 4 shows that after 10 rounds (note that only the gold standard dataset is used in the first round), the total number of samples entering the training is 768 (48 + 9 × 16 × 5). After training with AL, 93% (0.68 / 0.73) of the model performance was achieved with a total of less than 42%

(768/ (2312 × 0.8)) of the data. Figure 3 points out the performance gain after each round of AL and the performance difference between the baseline model and the proposed model trained with all training data. This suggests that a training model with an AL procedure continuously improved the model's performance with a CbITS proposed dataset (multiple semantic categories and domain specific). Considering that this example included 16 categories, the probability of random guessing is only about 6.3% (even if one considers guessing by the category with the most samples, the probability would only be 16.4%), the performance of the model is intuitively quite satisfactory.

Table 3 Model's performance when training with whole training data. Comparisons were conducted among the baseline model and different extra-layers structure

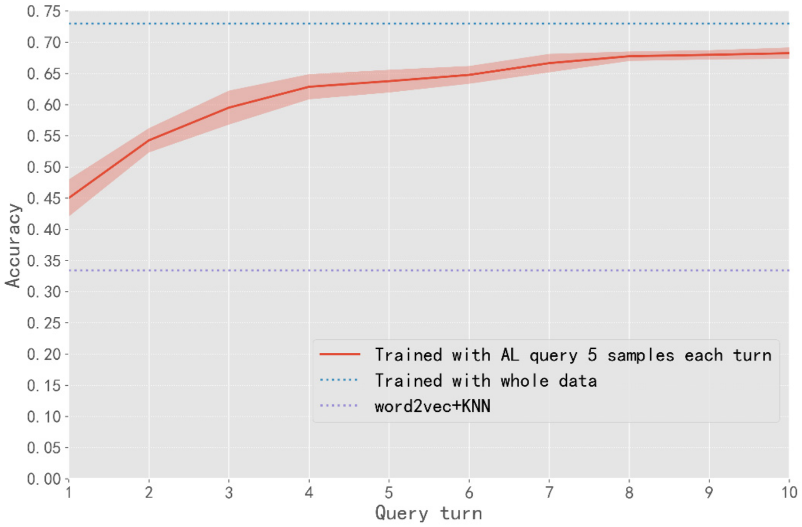
	Precision	*Recall*	*f1*	*Accuracy*
word2vec+knn	0.333	0.370	0.285	0.334
BERT+mean-pool	**0.681**	0.637	0.638	0.697
BERT+max-pool	0.659	0.561	0.569	0.641
BERT+attention	0.678	0.660	**0.662**	**0.732**
BERT+bi-LSTM+attention	0.636	**0.670**	0.635	0.713

Note: *Bold numbers mean maximum values.

Table 4 Training model with AL, with each round including 5 samples per semantic category samples queried

Round	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
precision	0.393	0.472	0.541	0.609	0.610	0.616	0.622	0.633	0.640	0.633
recall	0.457	0.450	0.526	0.568	0.578	0.592	0.603	0.620	0.621	0.623
f1	0.390	0.422	0.503	0.558	0.567	0.583	0.593	0.609	0.611	0.611
accuracy	0.450	0.543	0.595	0.628	0.638	0.648	0.666	0.677	0.680	0.682

Figure 3 Performance comparison between models trained with AL. The red area around the line shows the standard deviation during training



When training with deep learning models, the greater the amount of data, the better the outcome. With AL, the greater the number of samples queried per round, the better the performance. However, in practical applications, as one increases the number of queried samples, the workload of the human annotators increases proportionately. Therefore, it becomes important to determine the degree to which the model performance decreases when fewer samples are queried. We tested the performance of the model in each round when each semantic category query varied, from 1, 3, 5, and 10 samples, respectively. Table 5 shows the performance gap under different conditions, the baseline model, and the model trained with all of the data. Here we see that the greater the number of samples in each round of queries, the better the model performs. However, a smaller number of queries can also lead to some significant performance gains (all significantly higher than the baseline model): for example, just 1 query sample per semantic category per round (only 8% of total sample is used) achieved an effect of 79% (0.579/0.73). When the practicality of an application is of prime importance, one may need to consider the effect of the model and the load on the human annotator when deciding the optimal number of query samples to include.

Table 5 Model's performance under a different number of queries for each round (indicated by the first column). Numbers in table are accuracy scores for each test set

#queries	*Round 1*	*Round 2*	*Round 3*	*Round 4*	*Round 5*	*Round 6*	*Round 7*	*Round 8*	*Round 9*	*Round 10*
1	0.450	0.479	0.510	0.513	0.525	0.545	0.540	0.569	0.578	0.589
3	0.446	0.526	0.555	0.583	0.610	0.624	0.640	0.641	0.645	0.657
5	0.450	0.543	0.595	0.628	0.638	0.648	0.666	0.677	0.680	0.682
10	0.453	0.547	0.601	0.668	0.670	0.677	0.686	0.684	0.697	0.711

5.5 Evaluating in application scenario

We provide another simulation application scenario here for illustrative purposes. Consider the case where a gold standard dataset exists, with an unprocessed collection of sentences. Assume further the existence of a field expert (say, a junior in medical school) acting as a supervisor to classify the queries of active learning. Other settings are the same as described in the previous experiments. In this experiment, we compared the results with those obtained in the previous experiments. As repeated manual labelling is labour intensive, we reduced the number of query rounds to 5 (5 samples queried every round) and 5 times. The main findings, shown in Table 6, show that using preprocessed data led to better performance.

Table 6 Comparison between model trained with preprocessed dataset and raw dataset

Query round	*With preprocessed data*				*With raw data*			
	precision	*recall*	*f1*	*accuracy*	*precision*	*recall*	*f1*	*accuracy*
1	0.393	0.457	0.390	0.450	0.393	0.463	0.388	0.452
2	0.472	0.450	0.422	0.543	0.435	0.483	0.419	0.507
3	0.541	0.526	0.503	0.595	0.464	0.502	0.443	0.536
4	0.609	0.568	0.558	0.628	0.499	0.507	0.467	0.563
5	0.610	0.578	0.567	0.638	0.513	0.518	0.478	0.576
6	0.616	0.592	0.583	0.648	0.531	0.542	0.495	0.584

6 Discussion and prospect

6.1 Discussion

Natural language processing is experiencing a period of rapid development. However, the application of corresponding technologies in ITS, especially with respect to dialogue-based ITS, lags behind. In this paper, we proposed using deep learning models to perform semantic matching in conversation-based ITS and introduced active learning for alleviating the problem of labelling large amounts of data. Although related techniques (deep learning models and active learning) have been shown to be effective, we were unable to find a study testing their effectiveness in the field of learning, the prominent feature of which is the high degree of domain specificity. We believe that the main contribution of this paper is to evaluate the effectiveness of this method with data that are more in line with the learning field and to clarify some key problems when applying the proposed method.

Based on the evaluations reported herein, the proposed model was shown to greatly improve the accuracy of semantic matching: accuracy increased from 33% for the baseline model to over 60% (dependent upon details of implementation, like query rounds and number of queries each round), while needing very small amounts of labelled data. By querying more samples, each round served to increase the model's performance; however, it also increased the human supervisor's workload. Our results showed that even when a small number of samples were queried, this was sufficient for the model to surpass the performance level obtained with the baseline model. In other words, a small amount of extra effort (annotating queried samples in this context) was able to significantly improve CbITS's understanding of learners' natural language input. We believe this to be the core contribution of this study – showing that this scheme has practical application value.

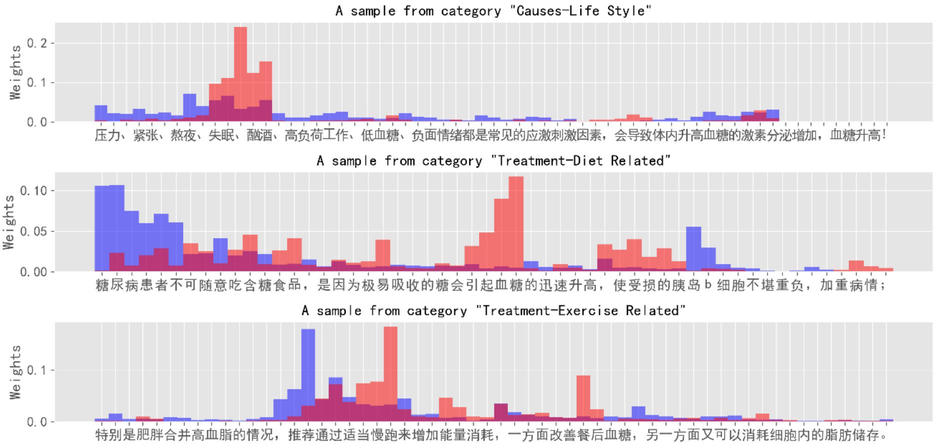
It is important to note that in the first round of testing, where the proposed model was trained only with the gold standard dataset, the model's performance was significantly better than baseline (in accuracy: 33% vs. 45%). Further, once annotated samples were used performance of the model continued to increase. This held true even if only 10 or more samples were used at a time and the sample was the original text without preprocessing. In practice, test sets may not exist, making it impossible to know when the performance of the model is close to its limit or when it is prudent to discontinue training. Lacking additional data, our experiments indicate for now that as long as the chosen method continues to run, benefits may be expected to accrue.

Preparing a training dataset is also often labour intensive, because it involves selecting the samples believed to be most useful from a large number of sentences. When we trained the model with AL under a condition where no preprocessing was conducted, we found that performance of the model was lower. Although using a set of raw un-preprocessed sentences was able to achieve acceptable results (significantly outperform the baseline model), dealing with a large set of sentences greatly increases the running time. Therefore, we recommend that a simple semantic similarity calculation be performed as described herein to build a training data set capable of filtering out sentences that are too dissimilar to the gold standard dataset provided by the domain expert. In many cases, developers of CbITS courses may not have experience in NLP-related fields, but they can easily come up with expectations and misconceptions related to the learning content. Our experiments demonstrate that while not the optimal solution, one needs only to have supported material (domain-related corpus) along with some labelled samples to be able to train a semantic understanding module with acceptable performance (i.e., one that is likely to be much better than the baseline model currently in use) to build an adaptive CbITS. This type of approach makes sense at the moment, because it requires a great deal of time and effort to create an efficient module to perform natural language processing for CbITS.

Our experimental results showed that our proposed method is effective, both in training a deep learning text classifier with fewer data and in distinguishing texts in a domain where similar texts may contain different point of views. Whether our experimental results can be extended to other domains remains unknown (with this being an area in need of further research). We believe that our approach is extendable, for two chief reasons: First, the pre-training language model performed well in other more complex tasks, and we perceive it is competent enough for text classification tasks. Second, the domain (basic diabetes-related knowledge) we chose for demonstration is not particularly special, and we are unaware of any evidence suggesting that this domain would make the text classification task easier to perform than others we may have selected. However, we acknowledge that additional evidence is needed to support our claim about the extendibility of our approach.

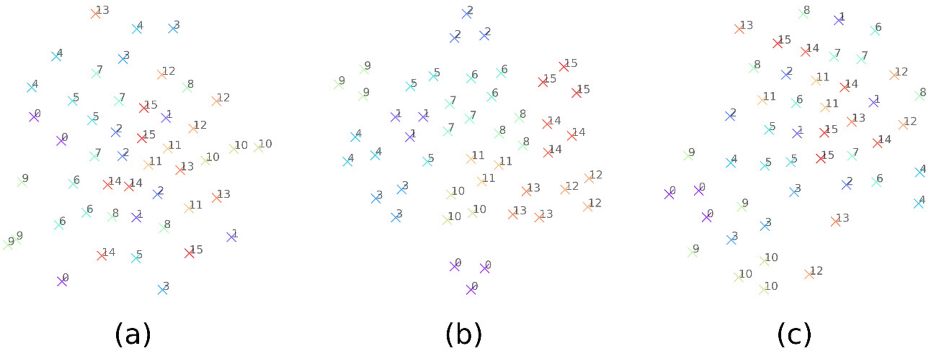
When dealing with natural language, which is typically unstructured data, the main advantage of deep learning models over traditional machine learning models is their ability to engage in representation learning (Bengio et al., 2013). Such models can learn a good representation of objects through training. Although no comparisons were made to traditional machine learning models, we provide two examples to illustrate this advantage with respect to deep learning models. We sampled some sentences and calculated the attention value (i.e., the output of the model’s attention layer, which indicates the importance of the corresponding words) of each word, as shown in Figure 4. The model we describe herein had learned to focus on more critical words for classification, which suggests that during training, the model captured the most informative cues for the task.

Figure 4 Learned attention weights for words/characters of samples. The model learned to focus on ’informative’ words that help determine if a pair of sentences belongs to the same semantic category. Note that the blue part shows the result of the 1st round AL of the model, and the red part shows the result of the 10th round



Moreover, Figure 5 visualises (using T-SNE; van der Maaten and Hinton, 2008) the results of entire samples in the gold standard datasets through different encoding methods. This includes the average on PTLM's output (shown in sub-figure a), output of the proposed model's encoder module (shown in sub-figure b), and average on word2vec vectors (shown in sub-figure c). Further, Figure 5b shows a good clustering pattern (i.e., the samples that belong to the same semantic category are relatively close), while the remaining clustering patterns (5-a and 5-c) are more chaotic, again demonstrating the advantages of using deep learning models, which are able to learn better ways to represent the domain of interest.

Figure 5 Visualisation of the golden standard dataset with different encoding methods: (a) mean pooling on BERT's output, (b) output of proposed model's encoder and (c) mean pooling (average) on word2vec vectors



6.2 *Prospect*

Although deep learning is very effective, two issues merit continued attention. First, in a smaller granularity, does deep learning make a difference in certain contexts, such as when students answer a question in natural language? Can it effectively detect the wrong description of a knowledge point? Currently, in CbITS, the typical way to solve this problem is to use pattern matching, such as in Latham et al. (2012). Second, at present, neither machine learning/deep learning nor pattern matching can correctly evaluate learners' natural language at 100%. For example, our findings show that although judging the semantic category of a sentence is a relatively simple task (no complex knowledge or reasoning is involved), the deep learning algorithm was able to achieve only a maximum accuracy of about 70%. In other words, with the current technical means, the phenomenon of misunderstanding of learners' input in CbITS may not be eliminated. So how should CbITS developers deal with this? Answers to these and related questions merit further exploration by future CbITS designers and developers.

Computing power and the absence of a large amount of high-quality data are two key factors that hinder the large-scale applications of deep learning models. The proposed model's success relies chiefly on the strong representation ability of a deep learning model. Our research approach is designed to reduce the need for data in model training; however, the need for a certain level of computing power remains. This may, at present, restrict large-scale application of the system as presented here.

Although we were able to demonstrate that using more advanced deep learning techniques could help improve natural language understanding in CbITS contexts, we have not yet arrived at the optimal approach. A missing key to fully addressing the present puzzle is our inability to understand semantics at a finer granularity. For example, if we asked about the causes of diabetes, a student may say, "Lack of insulin leads to low levels of blood sugar." The model we describe above may classify this answer as matching the expected category; however, the key information in this answer is incorrect (the correct answer is "high level blood sugar" not "low level"). A CbITS should be able to detect this kind of error and provide corresponding feedback. At present, the main solutions to this problem rely on pattern matching (for example Khuwaja et al., 1994) or deep syntactic analysis-based reasoning, which either places too many restrictions on the answer (string matching requires a few words of answer length) or is difficult to develop (for example in Popescu, 2005).

7 Conclusion

Our findings reported herein lead us to conclude that combining the deep learning model (mainly the pre-training language model) with AL is an improved scheme for natural language understanding in CbITS. By doing so, we can not only make use of the excellent semantic representation ability of deep learning, but also avoid expending excessive effort on collecting training data. With respect to research question 1, in a dataset closer to a given learning field, our proposed scheme significantly outperformed the baseline model (currently in use). Regarding research question 2, the application of our method for enhancing performance required that we only needed to make a minimal effort. For example, providing only a gold standard dataset (which contained only a very small number of samples per semantic category) allowed us to achieve a 36% performance improvement. If a domain text set is provided, along with labelling for a small number of samples, the performance improvement was found to be more significant (and may well exceed over 100%). In summary, we believe that our research provides practical guidance for improving natural language understanding in CbITS.

Acknowledgements

This publication resulted (in part) from research supported by the National Science Foundation of China (NSFC, #61937001). The content is solely the responsibility of the authors and does not necessarily represent the official views of NSFC.

References

20 Newsgroups. (n.d.) Available online at: http://qwone.com/~jason/20Newsgroups/ (accessed on 31 December 2020).

Aleven, V.A. and Popescu, O. (2003) 'A tutorial dialog system to support self-explanation: evaluation and open questions', *Proceedings of the 11th International Conference on Artificial Intelligence in Education 2003*, pp.39–46.

An, B., Wu, W. and Han, H. (2018) 'Deep active learning for text classification', *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, pp.1–6.

Arora, S., Liang, Y. and Ma, T. (2016) 'A simple but tough-to-beat baseline for sentence embeddings', *Proceedings of the 5th International Conference on Learning Representations, ICLR, 2017*, 24–26 April, Toulon, France.

Bahdanau, D., Cho, K. and Bengio, Y. (2014) 'Neural Machine Translation by Jointly Learning to Align and Translate', In *arXiv* [cs.CL].

Barlow, H.B. (1989) 'Unsupervised learning', *Neural Computation*, Vol. 1, No. 3, pp.295–311.

Bengio, Y., Courville, A. and Vincent, P. (2013) 'Representation learning: a review and new perspectives', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp.1798–1828.

Chen, Q., Zhuo, Z. and Wang, W. (2019) 'BERT for Joint Intent Classification and Slot Filling', *ArXiv*, abs/1902.10909.

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014) 'Empirical evaluation of gated recurrent neural networks on sequence modeling', *ArXiv*:1412.3555v1.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', *arXiv*:1810.04805v2.

Dumais, S.T. (2013) 'Latent semantic analysis', *Annual Review of Information Science & Technology*, Vol. 4, No. 6, pp.683–692.

Dzikovska, M., Bental, D., Moore, J., Steinhauser, N., Campbell, G.E., Farrow, E. and Callaway, C. (2010) 'Intelligent Tutoring with Natural Language Support in the Beetle II System', *Proceedings of the 5th European Conference on Technology-enhanced learning conference on Sustaining TEL: from innovation to learning and practice*.

Evens, M.W., Lee, Y.H., Shim, L.S., Chong, W.W. and Rovick, A.A. (2002) 'CIRCSIM-Tutor: An intelligent tutoring system using natural language dialogue', *ANLP*.

Glass, M. (2001) 'Processing language input in the CIRCSIM-Tutor intelligent tutoring system', in Moore, J.D. et al. (Eds): *Artificial Intelligence in Education*, IOS Press, pp.210–221.

Graesser, A.C., Chipman, P., Haynes, B.C. and Olney, A. (2005) 'AutoTutor: an intelligent tutoring system with mixed-initiative dialogue', *IEEE Transactions on Education*, Vol. 48, No. 4, pp.612–618.

Graesser, A.C., D'Mello, S., Hu, X., Cai, Z., Olney, A. and Morgan, B. (2012) 'AutoTutor', *Applied Natural Language Processing: Identification, Investigation and Resolution*, IGI Global, pp.169–187.

Graesser, A.C., Hu, X. and McNamara, D.S. (2005) 'Computerized learning environments that incorporate research in discourse psychology, cognitive science, and computational linguistics', in Healy, A.F. (Ed.): *Experimental Cognitive Psychology and its Applications: Festschrift in Honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer*, American Psychological Association, Washington, D.C., pp.183–194.

Graesser, A.C., Lu, S., Jackson, G.T., Mitchell, H.H., Ventura, M., Olney, A. and Louwerse, M.M. (2004) 'AutoTutor: a tutor with dialogue in natural language', *Behavior Research Methods Instruments & Computers*, Vol. 36, No. 2, pp.180–192.

Halpern, D.F., Millis, K., Graesser, A.C., Butler, H., Forsyth, C. and Cai, Z. (2012) 'Operation ARA: a computerized learning game that teaches critical thinking and scientific reasoning', *Thinking Skills & Creativity*, Vol. 7, No. 2, pp.93–100.

Hu, R., Mac Namee, B. and Delany, S.J. (2016) 'Active learning for text classification with reusability', *Expert Systems with Applications*, Vol. 45, pp.438–449.

Hu, X., Cai, Z., Wiemer-Hasting, P., Graesser, A. and McNamara, D.S. (2007) 'Strengths, limitations, and extensions of LSA', in Landauer, T., McNamara, D.S., Dennis, S. and Kintsch, W. (Eds): *Handbook of Latent Semantic Analysis*, Erlbaum, Mahwah, NJ, pp.401–426

Kalliopi-Irini, M., Wiemer-Hastings, P. and Robertson, J. (2002) 'Beyond the short answer question with research methods tutor', *International Conference on Intelligent Tutoring Systems*, pp.562–573.

Khuwaja, R.A., Evens, M.W., Michael, J.A. and Rovick, A.A. (1994) 'Architecture of CIRCSIM-Tutor (v.3): A Smart Cardiovascular Physiology Tutor', *Proceedings of the 7th Annual IEEE Computer-Based Medical Systems Symposium*, Winston-Salem, NC, IEEE Computer Society Press, pp.158–163.

Kim, Y. (2014) 'Convolutional neural networks for sentence classification', *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 25–29 October, Doha, Qatar, pp.1746–1751.

Latham, A., Crockett, K., McLean, D. and Edmonds, B. (2012) 'Adaptive tutoring in an intelligent conversational agent system', in Nguyen, N.-T. (Ed.): *Transactions on Computational Collective Intelligence VIII*, Springer, Berlin Heidelberg, pp.148–167.

Litman, D.J. and Silliman, S. (2004) *ITSPOKE: An Intelligent Tutoring Spoken Dialogue System*, NAACL.

Lu, J. and MacNamee, B. (2020) 'Investigating the Effectiveness of Representations Based on Pretrained Transformer-based Language Models in Active Learning for Labelling Text Datasets', *ArXiv*, abs/2004.13138.

Matthews, D., Vanlehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A. and Andrew, R.A. (2010) 'When are tutorial dialogues more effective than reading?' *Cognitive Science*, Vol. 31, No. 1, pp.3–62.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013) 'Efficient Estimation of Word Representations in Vector Space', *arXiv* [cs.CL]. http://arxiv.org/abs/1301.3781

Miller, B., Linder, F. and Mebane, W.R. (2020) 'Active learning approaches for labeling text: review and assessment of the performance of active learning approaches', *Political Analysis: An Annual Publication of the Methodology Section of the American Political Science Association*, Vol. 28, No. 4, pp.532–551.

Millis, K., Forsyth, C., Butler, H., Wallace, P., Graesser, A. and Halpern, D. (2011) 'Operation ARIES!: A Serious Game for Teaching Scientific Inquiry. serious games & edutainment applications', *Serious Games and Edutainment Applications*, Springer, London, pp.169–195.

Nowak, J., Taspinar, A. and Scherer, R. (2017) 'LSTM recurrent neural networks for short text and sentiment classification', *Artificial Intelligence and Soft Computing*, pp.553–562.

Pang, B. and Lee, L. (2004) 'A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts', *Proceedings of the 42nd ACL*, *arXiv*:cs/0409058v1, pp.271–278.

Pennington, J., Socher, R. and Manning, C.D. (2014) 'GloVE: Global Vectors for Word Representation', *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 25–29 October, Doha, Qatar, pp.1532–1543.

Person, N.K., Graesser, A.C., Kreuz, R.J., Pomeroy, V. and TRG (2001) 'Simulating human tutor dialog moves in AutoTutor', *International Journal of Artificial Intelligence in Education*, Vol. 12, pp.23–39.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018) 'Deep contextualized word representations', *arXiv*:1802.05365v2.

Popescu, O. (2005) *Logic-based natural language understanding in intelligent tutoring systems*, PhD Thesis, Carnegie Mellon University.

Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I. (2018) *Improving language understanding by generative pre-training*, cs.ubc.ca.

Rietzler, A., Stabinger, S., Opitz, P. and Engl, S. (2019) 'Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification', arXiv preprint *arXiv*:1908.11860.

Ritter, S., Anderson, J.R., Koedinger, K.R. and Corbett, A. (2007) 'Cognitive Tutor: applied research in mathematics education', *Psychon Bull Rev*, Vol. 14, No. 2, pp.249–255.

Rosé, C.P. and Vanlehn, K. (2005) 'An evaluation of a hybrid language understanding approach for robust selection of tutoring goals', *International Journal of Artificial Intelligence in Education*, Vol. 15, No. 4, pp.325–355.

Rücklé, A., Eger, S., Peyrard, M. and Gurevych, I. (2018) 'Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations', arXiv preprint *arXiv*:1803.01400, 2018.

Settles, B. (2009) *Active learning literature survey*, Available online at: http://digital.library.wisc.edu/1793/60660

Sun, C., Qiu, X., Xu, Y. and Huang, X. (2019) 'How to Fine-Tune BERT for Text Classification?' in Sun, M., Huang, X., Ji, H., Liu, Z. and Liu, Y. (Eds): *Chinese Computational Linguistics. CCL 2019. Lecture Notes in Computer Science*, Vol. 11856, Springer, Cham. https://doi.org/10.1007/978-3-030-32381-3_16

Sun, L.-L. and Wang, X.-Z. (2010) 'A survey on active learning strategy', *Proceedings of the 2010 International Conference on Machine Learning and Cybernetics*, Qingdao, China. DOI: 10.1109/ICMLC.2010.5581075.

Tong, S. and Koller, D. (2001) 'Support vector machine active learning with applications to text classification', *Journal of Machine Learning Research*, Vol. 2, pp.45–66.

van der Maaten, L. and Hinton, G. (2008) 'Visualizing data using t-SNE', *Journal of Machine Learning Research*, Vol. 9, No. 86, pp.2579–2605.

Vanlehn, K. (2011) 'The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems', *Educational Psychologist*, Vol. 46, No. 4, pp.197–221.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł.U. and Polosukhin, I. (2017) 'Attention is All you Need', *arXiv*:1706.03762v5 [cs.CL].

Yang, W., Zhang, H. and Lin, J. (2019) 'Simple Applications of BERT for Ad Hoc Document Retrieval', *arXiv* [cs.IR]. http://arxiv.org/abs/1903.10972

Zhang, Y., Lease, M. and Wallace, B.C. (2016) 'Active discriminative text representation learning', arXiv Preprint *arXiv*:1606.04212.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H. and Xu, B. (2016) 'Attention-based bidirectional long short-term memory networks for relation classification', *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 2: Short Papers, pp.207–212.

Zhu, J. and Hovy, E. (2007) 'Active learning for word sense disambiguation with methods for addressing the class imbalance problem', *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp.783–790.

Zhu, J., Wang, H., Yao, T. and Tsou, B.K. (2008) 'Active learning with sampling by uncertainty and density for word sense disambiguation and text classification', *Proceedings of the 22nd International Conference on Computational* Linguistics, pp.1137–1144.

Appendix

Samples of categories in Table 1. The original samples are in Chinese, but the text reported here contains the English translations for demonstration purposes.

Category	*Sub-category*	*Sample*
Diabetes	Definition	In type 2 diabetes, the body can produce insulin, but due to a relative lack of insulin secretion or deficiency of action (also known as insulin resistance), blood sugar increases.
	Description	Type 2 diabetes is a very common metabolic disease.
	Categories	Clinically, diabetes is mainly divided into two types, type 1 diabetes and type 2 diabetes.
	Clinical symptoms	Clinically, hyperglycaemia is the main feature, and typical cases may have symptoms such as polyuria, polydipsia, polyphagia and so on.
Epidemiological characteristics		China's national health big data shows that China's total population is 1.4 billion, there are 140 million diabetic patients, and one and four diabetics in the world are all Chinese.
Treatment	Hypoglycae mic drugs	The hypoglycaemic mechanism of sulfonylureas is mainly to stimulate insulin secretion, which is suitable for patients with complete islet function. Obese patients should be combined with weight control and biguanide hypoglycaemic drugs.
	Insulin related	The hypoglycaemic effect of quick-acting insulin analogues is similar to that of short-acting human insulin, but superior to human insulin in simulating physiological insulin secretion, reducing the amplitude of PPG and the risk of hypoglycaemia.
	Diet related	People with diabetes should not eat sugary foods at will, because the sugars that are easily absorbed will cause a rapid increase in blood sugar, which will overwhelm the damaged islet B cells and aggravate the disease.
	Exercise related	Exercise has an important therapeutic effect that cannot be replaced by drugs. Regular and effective moderate intensity exercise therapy can significantly reduce the level of blood glucose in patients with type 2 diabetes mellitus.
	General	Timely and correct treatment is very helpful to patients with type 2 diabetes.
Cause of disease	Life style	Stress, tension, staying up late, insomnia, alcoholism, work difficulties, hypoglycaemia and negative emotions are all common stress stimuli, which may lead to increased hormone secretion, thereby raising blood sugar levels.
	Other diseases	More and more scientific studies have shown that children who don't get enough sleep are more likely to develop type 2 diabetes.
	Heredity	The heritability of type 1 diabetes is about 72-88%, that is to say, congenital genetic factors have more influence than acquired environmental factors.

Category	*Sub-category*	*Sample*
Harm	Harm	As these tissues and organs are forced to operate in a state of hyperglycaemia, over time, there will naturally be the tragedy of damage, dysfunction, failure, and eventually lead to complications.
Diagnosis	Diagnosis	The fasting blood glucose of normal people was less than 6.1 mmol/L, and after taking glucose for 2 hours, the blood glucose was less than 7.8 mmol/L.
Related metabolism	Related metabolism	The control of blood sugar level in the body needs the pancreas to play a role. The pancreas has islet A cells and islet B cells. A cell is responsible for raising blood sugar and secretes a hormone called glucagon.