User Personalized Satisfaction Prediction via Multiple Instance Deep Learning

Zheqian Chen
State Key Lab of CAD&CG
Zhejiang University
zheqianchen@gmail.com

Ben Gao
State Key Lab of CAD&CG
Zhejiang University
beng.zju@gmail.com

Huimin Zhang
State Key Lab of CAD&CG
Zhejiang University
rachelbowlong@gmail.com

Zhou Zhao
College of Computer Science
Zhejiang University
zhaozhou@zju.edu.cn

Haifeng Liu
College of Computer Science
Zhejiang University
haifengliu@zju.edu.cn

Deng Cai
State Key Lab of CAD&CG
Zhejiang University
dengcai@cad.zju.edu.cn

ABSTRACT

Community question answering(CQA) services have arisen as a popular knowledge sharing pattern for netizens. With abundant interactions among users, individuals are capable of obtaining satisfactory information. However, it is not effective for users to attain satisfying answers within minutes. Users have to check the progress over time until the appropriate answers submitted. We address this problem as a user personalized satisfaction prediction task. Existing methods usually exploit manual feature selection. It is not desirable as it requires careful design and is labor intensive. In this paper, we settle this issue by developing a new multiple instance deep learning framework. Specifically, in our settings, each question follows a multiple instance learning assumption, where its obtained answers can be regarded as instance sets in a bag and we define the question resolved with at least one satisfactory answer. We design an efficient framework exploiting multiple instance learning property with deep learning tactic to model the question-answer pairs relevance and rank the asker's satisfaction possibility. Extensive experiments on large-scale datasets from different forums of Stack Exchange demonstrate the feasibility of our proposed framework in predicting asker personalized satisfaction.

Keywords

User Satisfaction Prediction; Multiple Instance Learning; Deep Learning

1. INTRODUCTION

Community-based question answering(CQA) services have emerged as prevalent and helpful platforms to share knowledge and to seek information for netizens. With abundant interactions and fully openness, CQA services enable users to directly obtain specific information from other community participants. However, there is a fundamental problem in CQA services. Users may take days or even weeks to wait for a satisfactory answer posted. It is too time-consuming and users may not have so much patience to check the progress and get the question resolved. Hence how to predict the user's personalized satisfaction have become inevitably crucial. In this paper, we target at predicting the user's individual satisfaction possibility. It is meaningful to resolve this challenge, CQA services can thus timely inform askers the results so that they do not have to check the progress overtime.

Nevertheless, predicting user satisfaction in QA community is challenging since satisfaction is inherently subjective for askers. It is impractical to directly regard the most semantic relevant answer as the satisfactory one in QA pairs since users preferences vary from person to person. Although this matching ranking method is the mainstream in question answering field to recommend best answers. Majority of existing studies in the user satisfaction prediction task adopt feature engineering methods and cast this problem as a binary classification task [13] [14] [15]. They typically employ manual feature selection and apply supervised machine learning algorithms on these features. Indubitably feature engineering achieves considerable progress, but it is labor intensive and requires cautious design.

To avoid complicated feature engineering, how can we extract and organize discriminative features automatically from data? As the superior performance of deep learning, an intuitive idea is to combine deep learning method to replace manual feature extraction. Moreover, we observe that generally in CQA portals, answers usually come with high diversity but much noise. Users may not assign which answer is the most satisfied, but just close the question as have satisfied with the whole answer states. Under this assumption, we realize that this property actually is applicable to the assumption in multiple instance learning, which indicates that each positive bag must have at least one positive instance. Therefore, we attempt to absorb multiple instance learning into a deep learning framework to assist the task of user personalized satisfaction prediction. Specifically, in our settings, a question with several answers can be treated as a bag with certain instances. We regard a question resolved with at least one satisfactory answer, which is the same as one positive bag contains at least one positive instance. To

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052599

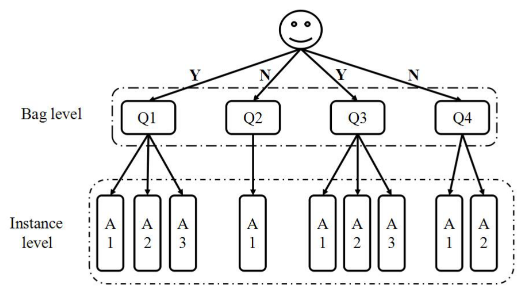

Figure 1: We denote each question in the bag level and each answer in the instance level. For a question asked by a user, we only know if the question has been assigned satisfied or not, while we don't know exactly which specific answer is assigned. This situation suits to the assumption of multiple instance learning. Here note Y means satisfied and N unsatisfied.

this end, we integrate user modeling [22] and recurrent neural network [9] with neural tensor network [19] to solve the multiple instance learning task, and introduce a **M**ultiple **I**nstance **D**eep **L**earning (**MIDL**) framework to effectively incorporate users' preferences and the question-answer pairs relevance. The general idea can be illustrated as Figure 1.

As is shown, in our settings, we do not need to know which answer will be evaluated as the satisfied one, what we need is the user's reaction to the whole answers on the basis of the question. This can be naturally modeled as multiple instance learning if we consider each answer as an instance and the answers for a question as a bag. Also we can say this situation is one of the weakly supervised learning patterns.

We conduct experiments to evaluate the effectiveness of the proposed method for the user personalized satisfaction prediction task. The source dataset we process is dumped from StackExchange website. Extensive experimental results show that our assumption of integrating multiple instance learning with deep learning outperforms several strong baseline methods which only use manual feature extraction. Moreover, considering the user's personalized preference shed light on improving effectiveness than just rank the question-answer pairs relevance.

It is worthwhile to highlight several contributions of our work here:

- We incorporate deep learning into a multiple instance learning framework named **MIDL** in a principled manner, where we put forward a new assumption in dealing with user personalized satisfaction prediction problem.
- Unlike previous studies, our proposed framework which leverages the multiple instance learning assumption and deep learning approach can be processed into an end-to-end procedure. Our framework can be extended into other weakly supervised learning scenarios.
- Our proposed framework achieves convincing performance than the state-of-the-art models which utilized manual feature extraction. The performance improved significantly, which demonstrate the potential of our concept of merging multiple instance learning with deep learning.

The remainder of this paper is organized as follows. In Section 2, we present a brief view of current related work about user personalized satisfaction prediction and deep learning with multiple instance learning. In Section 3, we formulate the user satisfaction prediction problem and introduce our proposed framework. In Section 4, we describe the experimental settings and report a variety of results to verify the superiority of our model. Finally, we conclude the paper in Section5.

2. RELATED WORK

We briefly review the related work on predicting users personalized satisfaction and the early approaches in studying multiple instance learning as well as current neural tensor network work in this section.

2.1 Users Personalized Satisfaction Prediction

Community-based question answering field has attracted substantial researchers to develop various algorithms to better retrieve and extract high-quality relevant information among participants. In previous studies, CQA researchers mainly focus on ranking the answers relevance and diversity, and regard the best ranking answer as the most satisfied results [6], [29], [1], [30]. A significant difference between QA-pairs matching ranking and users personalized satisfaction prediction is the user's latent preference. From the user's perspective, subjective response to question formulation, related experts recommending, relevant and novel answers taste vary from person to person. There also exist some superior work on modeling user preferences [28], [22], [23]. User satisfaction researches are popular in information retrieval field but is scarce in CQA field. The most relevant work with our user satisfaction prediction task in CQA field was presented by Liu [15] in 2008. Liu et. directly studied the satisfaction from CQA information seeker perspective, they incorporated a variety of content, structure and community-focused features into a general prediction model. Latha [12] integrated the available indicators and explored automatic ranking without explictly asking users to assess. In information retrieval field, Liu [14] analyzed unique characteristic of web searcher satisfaction in three aspects: query clarity, query-to-question match, and answer quality. Hassan [7] performed a large scale clickthrough data to explicit judge the user's sequential satisfaction level in the entire search task. Wang [24] hypothesized that users' latent satisfaction in action-level influences the overall satisfaction and built a latent structural learning method with rich structured features. We note that these existing methods in predicting the user's satisfaction are mainly depend on artificial extraction characteristics. Although they may gain considerable results, it is too labor intensive. As the flourish of deep learning, it may shed light on this problem.

2.2 Multiple Instance Learning

We observe that most deep learning methods are applied in fully supervised settings. However, in our assumption, predicting the user's satisfaction reaction under the condition that each question followed with several unlabeled answers, is basically a weakly supervised problem. In multiple instance learning settings, a bag with several unlabeled instances is assigned positive if and only if it contains at least one positive instance. Since the emergence of multiple instance learning by drug activity prediction researchers in 1990s [4], a number of researches have gain significant improvements. For example, Andrew [2] introduced MI-SVM and miSVM respectively from the bag-level and the instance-level. Zhang [27] improved DD algorithm by combining EM method and achieved the best result in the musk molecular data at that time. Vezhnevets [21] introduced Semantic Texton Forest to address the task of learning a semantic segmentation using multiple instance learning. Recently, researchers began to incorporate deep representations with multiple instance learning to enhance the performance. Specifically, Wu [25] designed CNN feature extraction method to jointly exploit the object and annotation proposals in vision tasks including classification and image annotation. Kraus [11] also studied a new neural network architecture with multiple instance learning in order to classify and segment microscopy images using only whole image level annotations. Xu [25] adopted multiple instance learning framework in classification training with deep learning features for medical image analysis. Zhou [31] investigated the web index recommendation problem from a multiple intance view, they regarded the whole website as a bag and the linkpages in website as the corresponding instances. We note that in multiple instance learning field, rare researchers have exploit deep learning tactics into Natural Language Processing task. Thus we further attempt to extend the application into CQA field.

2.3 Neural Tensor Network

Previous models suffer from weak interaction between two entities in the vector space. To address this problem, Socher et al. [19] first introduced the neural tensor network to allow the entities and relations to interact multiplicatively. They successively applied the neural tensor network to solve the problem in typical Natural Language Processing field. Later after the first proposal, they [20] introduced a new recursive neural tensor network to remedy sentiment detection task. Neural tensor network out-performed other linear combination approaches significantly and raised much attention among researchers. Chen [3] studied the problem of learning new facts with semantic words. In CQA field, researchers also adopt the idea of neural tensor network. Xia [26] modeled document novelty with neural tensor network for search result diversification task, they automatically learned a nonlinear novelty function based on preliminary representations of a document and other candidate documents. Qiu [17] integrated Q-A pairs semantic matching with convolutional and pooling layers, and exploited neural tensor network to learn the matching metrics. In our paper, we integrate neural tensor network to link the relevance of the user's attitude towards to the question accompanied with answers.

3. MULTIPLE INSTANCE DEEP LEARNING

In this section, we present the framework of Multiple Instance Deep Learning (MIDL). We first introduce the task of user satisfaction prediction on community question answering that we are seeking to solve and frame our formulation. Then we present the details of learning textual contents of U-Q-A representations with Recurrent Neural Network. And then we provide conceptual settings of multiple instance learning with neural tensor network. Finally we describe the training process and corresponding algorithm in a heuristic way.

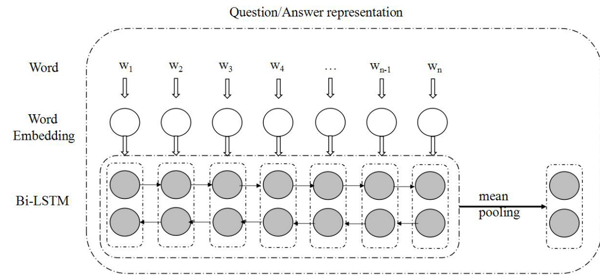

Figure 2: We adopt word embedding function and Bi-LSTM encoders to encode answers and questions. For Bi-LSTM encoders, we concatenate each unit hidden state within two layers and assign a mean pooling to get the global representation of the semantic content.

3.1 Task Description and Formulation

In this paper, we focus on predicting users personalized satisfaction. As is described earlier, in our formulation, we aware that it is reasonable to formulate that a user's satisfaction reaction lies in at least one of the corresponding satisfactory answers. In real world, when faced with a list of answers, users may have difficulties in deciding which answers are satisfied. However, it is justifiable to assume that the questions resolved with at least one satisfactory answer. In other words, it is natural to treat a question resolved as a positive bag with at least one of positive satisfactory answer instances. This property inspires us to design a multiple instance learning tactic to model the satisfaction prediction task.

Detailed manual annotations for each answer are time consuming for QA users. An alternative is to learn the global annotations for the overall answers, which is the main idea of multiple instance learning. Given the multiple instance learning assumption, questions with corresponding answers are organized as bags, which denotes as $\{\chi_i\}$. Within each bag there are a set of answer instances $\{\chi_{ij}\}$. We define the users satisfaction reactions as the labels $\{Y_i\} = \{1, -1\}$. In our proposal, The labels $\{Y_i\}$ are only available at the bag level, and we do not know the label at the instance level $\{y_{ij}\}$. The task is to predict the labels of unseen bags with multiple instances. We thus incorporate the multiple instance learning property into predicting the label of the user's satisfaction reaction at the bag level.

3.2 Modeling U-Q-A with Recurrent Neural Network

Considering the flourish of deep learning and the ideas of learning from data, an intuitive method is to combine deep learning method to replace manually feature extraction in learning the semantic embedding of questions and answers textual contents. In **MIDL** framework, we exploit Bi-directional LSTM for learning Q-A deep representations, which is inspired by [16]. The structure of our proposed Bi-directional LSTM is shown in Figure 2.

Intuitively, our framework of modeling U-Q-A semantic embedding is structured as follows:

1. We define a common user space and initialize the representation for each individual user in terms of their historical behaviors. For those who have rare record-

s we just assign the average representation from the whole corpus.

2. We embed each word to vector and apply Bi-directional LSTM to encode the contextual semantic representatins for questions and answers.

3. We concatenate the user representation with question embeddings and obtain the new semantic vectors of Q-U embedding.

In detail, we first construct representations for individual users corresponding to their historical behaviors. And then we employ word embedding function and Bi-directional LSTM encoder to encode User-specific-Question representation and Answer embedding into hidden vectors. We believe that using Bi-directional LSTM can better capture the contextual information from both directions as it can reduce the vanishing gradient problem. A Bi-directional LSTM consists of a forward LSTM and a backward LSTM. The forward LSTM reads each word w_i (i.e., from w_1 to w_i) in sequence as it is ordered, and generate the hidden states of each word as $\left(\overrightarrow{h_1}, ..., \overrightarrow{h_i}\right)$. For the backward LSTM, it processes each sentence in its reversed order (i.e., from w_i to w_1) and form a sequence of hidden states $\left(\overleftarrow{h_1}, ..., \overleftarrow{h_i}\right)$. We calculate the hidden states $\overrightarrow{h_i}$ by following equations:

$$\begin{aligned}
i_t &= \delta(W_i x_t + G_i h_{t-1} + b_i) \\
\hat{C}_t &= tanh(X_c x_t + G_f h_{t-1} + b_f) \\
f_t &= \delta(W_f x_t + G_f h_{h-1} + b_f) \\
C_t &= i_t \cdot \hat{C}_t + f_t \cdot C_t \\
o_t &= \delta(W_o x_t + G_o h_{t-1} + V_o C_t + b_o) \\
h_t &= o_t \cdot tanh(C_t)
\end{aligned}$$

where σ represents the sigmoid activation function; W_s, U_s and V_o are weight matrices; and b_s are bias vectors. There are three different gates (input, output, forget gates) for controlling memory cells and their visibility. The input gate can allow incoming signal to update the state of the memory cell or block it and the output gate can allow the state of the memory cell to have an effect on other neurons or prevent it. Moreover, the forget gate decides what information is going to be thrown away from the cell state.

Specifically, In our models, we first implement the word embedding function in a usual way, which exploit a look-up table and each word is indexed by one-hot representation from the vocabulary. We then adopt the popular mean pooling Bi-LSTM to encode the context. Since we care more about the relevance of each word in the text, we encode every word contextual embedding from Bi-LSTM unit and denote $\left\{h_{x,i} = \left[\overrightarrow{h_i}, \overleftarrow{h_i}\right]\right\}$ as the semantic embedding $f_i(a)$. And then we put a mean pooling layer to obtain the general semantic embedding for the original text.

3.3 Exploiting Multiple Instance Learning with Neural Tensor network

To model the user's attitude towards to the answers, we propose to use neural tensor network to measure the relationships between Q-U representation and the answers representations. Neural tensor network is proposed for reasoning over relationships between two entities [19]. Given two

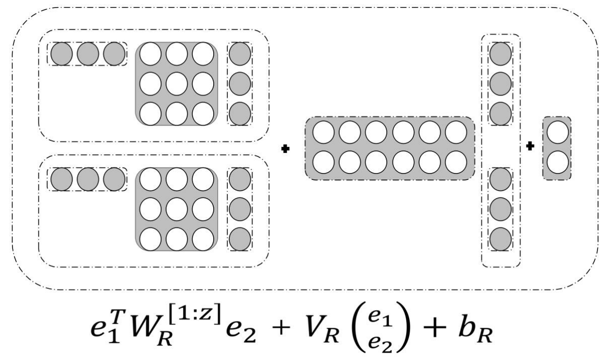

Figure 3: Visualization of the neural tensor network applied for entities relationships measurement.

entities (e_1, e_2) encoded with d dimensions, we use neural tensor network to state whether these two entities have a certain relationship R, and what the certainty is. We adopt the neural tensor network with a bilinear tensor layer to compute the relevance of two entity vectors across multiple dimensions. Assume $e_1, e_2 \in \mathbb{R}^d$ is the vector representations of the two entities, we compute the score of these two entities in a certain relationship. The equation is presented in the following:

$$g(e_1, R, e_2) = \mu_R^T tanh(e_1^T W_R^{[1:z]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R) \quad (1)$$

where $W_R^{[1:z]} \in \mathbb{R}^{d \times d \times z}$ is a tensor and we conduct the bilinear tensor product $e_1^T W_R^{[1:z]} e_2$ to gain a vector $h \in \mathbb{R}^d$. Each entry of h is computed by one slice $i = 1, ..., z$ of the tensor: $h_i = e_1^T W_R^{[1:z]} e_2$. The other parameters for relation R are the standard form of neural network: $V_R \in \mathbb{R}^{z \times 2d}$ and $U \in \mathbb{R}^z$, $b_R \in \mathbb{R}^z$. We reveal the original neural tensor network in Figure 3.

Intuitively, the origin neural tensor network is proposed to model the relationships between two entities with a bilinear tensor product. This conception can be naturally extended into modeling the relationships of a Q-U representation with respect to the answers representations. To this end, we adopt the neural tensor network into our multiple instance learning framework. The schematic diagram of our proposed framework is shown in Figure 4.

To learn multiple instances as a bag of samples, we incorporate the Q-U-A deep representations with multiple instance learning. We apply the modified version of neural tensor network to jointly learn the multiple instances within a bag. More specifically, assume that given the Q-U embedding $Q = \{d_i\}$ and the set of n answer embedding $A = \{d_j\}_{j=1}^n$. All of the embedding are obtained from preliminary Bi-directional LSTM representation. Given a Q-U representation $q \in Q$ and a set of answers $\{a_1, a_2, ..., a_n\}$. We extend the origin neural tensor network in the following equation:

$$g_n(q, A) = \mu^T max\left\{tanh(q^T W^{[1:z]} [a_1, a_2, ..., a_n])\right\} \quad (2)$$

We define the answers preliminary representation vectors $[a_1, a_2, ..., a_n]$ and form a matrix $M \in \mathbb{R}^{d \times n}$. $W_R^{[1:z]} \in$

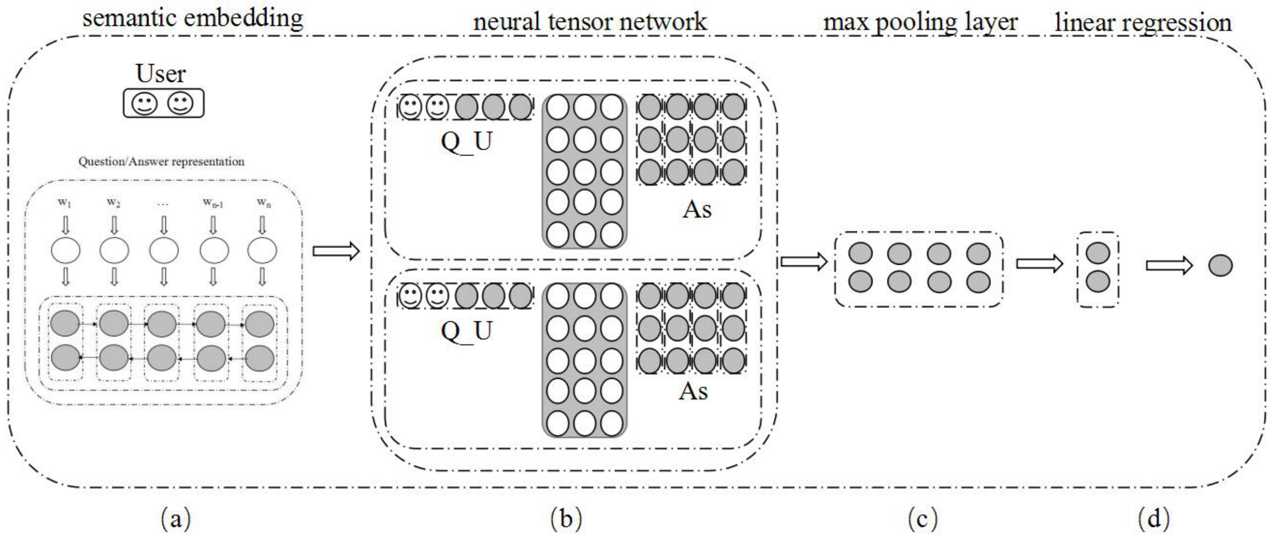

Figure 4: The overview of our proposed framework MIDL. (a)We adopt Bi-directional LSTM to learn the contextual content embedding of questions and answers, and initialize the user vector. (b) We concatenate the question embedding with user vector to form a Q-U representation. We then utilize the modified neural tensor network to model the relationships of Q-U representation and corresponding answers. (c)We put a max pooling layer to extract the most representative element, the pooling result represent the whole bag embedding for multiple instance learning. (d) The bag-level vectors are applied into logistic regression and obtain the predicting result of user satisfaction.

$\mathbb{R}^{d\times d\times z}$ is a tensor. For convenience we ignore the other bias term in original neural tensor network. We also conduct the bilinear tensor product $q^T W_R^{[1:z]}[a_1, a_2, ..., a_n]$ followed by a nonlinear operation:

$$H = \begin{bmatrix} h_1^T \\ ... \\ h_z^T \end{bmatrix} = \begin{bmatrix} tanh(q^T W^{[1]}[a_1, ..., a_n]) \\ ... \\ tanh(q^T W^{[z]}[a_1, ..., a_n]) \end{bmatrix} \quad (3)$$

Where $h_i \in \mathbb{R}^n$ is achieved by each slice of the tensor. Here we apply the hidden states H to model the user personalized attitude towards to a question with corresponding answers. The output of H is a matrix $z \times n$, in which each column is the representation of an instance. We aggregate the representation of the bag for multiple instance learning with max pooling:

$$v = \left[max(h_1^T), ..., max(h_n^T)\right]^T \quad (4)$$

Here we use max-pooling to extract the most significant element to well represent the whole bag for multiple instance learning. And finally, we adopt the bag representation v into a binary logistic regression which denotes “satisfied” or “not satisfied” to predict the label of the bag. Specifically, we formulate a binary multiple instance learning framework which optimized the loss function of bag classification. Denote $X_i = \{X_{i1}, X_{i2}, ..., X_{im}\}$ is the i^{th} bag of the question in the training set, and $\{X_{i1}, X_{i2}, ..., X_{im}\}$ is the answer instances. m is the number of answer instances in the i^{th} bag. $Y_i \in \{-1, +1\}$ is the label of the bag, 1 denotes “satisfied” and -1 denotes “not satisfied”. The loss function is:

$$L(H) = -\sum_{i=1}^{n} \mathbf{1}(Y_i = 1) log H(X_i) + \mathbf{1}(Y_i = -1) log(1 - H(X_i)) \quad (5)$$

Where $\mathbf{1}(\cdot)$ is an indicator function.

We iteratively train weak classifiers $h'(x)$ using gradient descent:

$$w_{ij} = \frac{\partial L(H)}{\partial h(x_{ij})} = -\frac{\partial L(H)}{\partial H(X_i)} \frac{\partial H(X_i)}{\partial h(x_{ij})} \quad (6)$$

where $h(x)$ updates by $h(x) + \alpha h'(x)$ and α is the parameter optimized by line searching. So far we generate an efficient classifier after the loss function converge.

3.4 Training

In this section, we present the details of our multiple instance deep learning **MIDL** method and summarize the main training process in Algorithm 1.

We begin with one-hot representations on each word, then we apply two Bi-directional encoders to denote questions and answers semantic representations respectively and we initialize the user embedding. After that, we concatenate each question with its asker to form the Q-U representation, which represents the asker's intent to the question. Afterwards we apply the Q-U representation with a set of answers to the updated neural tensor network to compute the relationships. And finally we use the logistic regression to predict the satisfaction level of users.

Denote all the parameters in our framework as Θ, we define the objective function in training process:

$$\underset{min}{\Theta} L(\Theta) = L(\Theta) + \lambda \left\|\Theta\right\|_2^2 \quad (7)$$

Algorithm 1 MIDL for Users Satisfaction Prediction

Require:

Input: Question-Answer Dataset $D(Q, A, U_{id})$,
question q, askerid uid, the i_{th} answers set of q is A_q

Pre-train the word-embedding of Q and A by skip-gram
Initialize the user embedding
for q in Q **do**
  for a in A_q **do**
    $a\_emb = lstm(a)$
  end for
  $q\_emb = lstm(q)$
  $u\_emb = U(uid)$
  neural-tensor(q_emb,a_emb,u_emb)
  Summate the total training loss
  Update parameters by SGD
end for

$\lambda > 0$ is a hyper-parameter to trade-off the training loss and regularization. By using SGD optimization with the diagonal variant of AdaGrad as in [5], at time step t, the parameter Θ is updated as follows:

$$\Theta_t = \Theta_{t-1} - \frac{\rho}{\sqrt{\sum_{i=1}^{t} g_i^2}} g_t \tag{8}$$

where ρ is the initial learning rate and g_t is the subgradient at time t.

4. EXPERIMENTS

To empirically evaluate and validate our proposed framework multiple instance deep learning(**MIDL**), we conduct experiments on a widely used dataset dumped from Stack Exchange community.

4.1 Data Preparation

The dataset downloaded from the famous community-based question answering portal Stack Exchange is an anonymized dump of all user-contributed content. The whole dataset consists of over 133 question answering forums and the S-tackOverFlow is the biggest forum among them. In our experiment, we snapshot four forums history data to validate our framework against some baselines. The theme of these four forums are “Android”, “Academia”, “Photo”, “Christian”. We present the detail of these four forums data in Table 1.

Table 1: Statistic of the four forums data

Forum	Question	Answer	User	Satisfied
Android	25310	42238	15845	42.1%
Academia	12062	31046	5875	50.6%
Photo	14414	38206	6867	59.6%
Christian	6915	17502	1777	53.9%

As we can see, questions in four forums received distinct proportion of answers, and the average satisfied ratio vary from each other. Among these four forums, the Android forum is the most popular but draws on only 1.67 answers for each question on average and the user’s satisfaction level is the lowest compared with other forums. The Photo forum get the highest satisfaction level with 59.6% and the most answers with 2.65 answers per question. In summary, asker satisfaction and other statistics of the questions vary widely from each forum data. We then split the dataset into training set, validation set and testing set without overlapping in our experiments. We fix the validation set as 10% of the total data to tune the hyperparameters and the size of testing set is 30%.

4.2 Evaluation Criteria

In order to evaluate the performance of different models, we employ Precision, Recall, F1-Measure and Accuracy as evaluation measures. These measure criterions are widely used in the evaluation for user satisfaction prediction task. Precision reports the ratio of the predicted satisfied question respect to the indeed rated satisfactory by users. Recall evaluates the fraction of all the indeed rated satisfactory questions that are distinguished by the framework. F1-Measure comprehensively analysis the results of Precision and Recall. Accuracy reflects the framework classification ability for the entire sample.

4.3 Performance Comparisons

To validate the performance of our approach, we compare our proposed method against with other eight state-of-the-art methods for the users personalized satisfaction prediction problem.

- **ASP_SVM** Support vector machines with manually selected features in [15]. In our experiment, we implement the relevant feature selection according to illustration in [15], and then we use libsvm to integrate the features to svm to classify the label of the user’s satisfaction result.
- **ASP_RF** RandomForest with manually selected features in [15]. Random forests are an ensemble method which was created by TK [8]. We use random forest classifier as well as feature selection in order to get high precision on the target label.
- **ASP_C4.5** C4.5 algorithm with manually selected features in [15]. C4.5 is used to generate a decision tree developed by JR Quinlan [18], and has become quite popular in classification. Here we use the same feature selection referred to [15].
- **ASP_Boost** Boosting algorithm with manually selected features in [15]. Boosting posed by Kearns [10] is primarily applied to reduce bias and variance in supervised learning, the idea of boosting is also from ensemble methodology.
- **ASP_NB** Naive Bayes with manually selected features in [15]. Naive Bayes classifier is based on applying Bayesian theorem with strong independence assumptions between the features, in this paper we also conduct the Naive Bayes classifier with selected features to fully evaluate the feasibility of our framework.
- **MISVM** MISVM Proposed by Andrews [2] is a classical multiple instance learning algorithm, it extend SVM to maximize the bag-level pattern margin over the hidden label variables. Here we address the predicting problem with MISVM to suit our settings.
- **EM-DD** Em-DD is a general-purpose for multiple instance problem that combines EM with the diverse

density(DD) algorithm [27]. We derive the idea of EM-DD algorithm to compare the performance with MIDL framework.

- **BP-MIP** BP-MIP [32] employs a specific error function derived from BP neural network. We implement the simplified version of BP-MIP to address our problem.

Overall, the first five classification baselines are supervised methods which focus on the feature selection manner and latter three are weakly supervised methods which are often applied in multiple instance learning. In order to better demonstrate the impact of different components of our proposed framework **MIDL**, we respectively evaluate the performance between manual feature selection and deep learning representations, and validate the feasibility of our assumption against with typical multiple instance learning algorithms.

In our experiments, we select the available features according to the reference of the paper [15]. we totally organized five basic entities in question answering community, which are questions, answers, Q-A pairs, users and categories. In summarize, we extract over 40 kinds of different features from the five entities. This process is quite labor-intensive but we managed to implement the thorough feature extraction from the available corpus. For the three typical multiple instance learning algorithm, we strictly follow the idea from the paper and adapt the model to suit our user satisfaction prediction assumption. For fair we implement these eight baselines under the same constraints, all the hyperparameters and parameters which achieve the best performance on the validation set are chosen to conduct the testing evaluation.

4.4 Experimental Results and Analysis

To evaluate the performance of our proposed framework, we conduct several experiments on four metrics described above.

Table 1, 2, 3 and 4 show the evaluation results on Precision, Recall, F1-Measure and Accuracy, respectively. We conduct the experiments with four datasets extracted from Stack Exchange website. We then report several interesting analysis that we observed on the evaluation results.

As mentioned previously, we argue that users personalized satisfaction can be assumed as a multiple instance learning problem. In order to verify our hypothesis, we conduct eight baselines trained with the same dataset and tested under the same evaluation criteria. Table 2, 3, 4 and 5 show the evaluation results in terms of four typical evaluation criterias Precision, Recall, F1-Measure and Accuracy. Figure 5 explores the tendency of performance with varying amount of training data in our framework. Figure 6 shows the prediction accuracy for distinct groups of users with different number of questions.

With these experimental results, we can summarize several interesting points:

- We observe that in most cases in four forum datasets our proposed framework MIDL outperforms other baselines significantly, which suggests that it is feasible for us to hypothesis the users personalized satisfaction prediction problem into multiple instance learning formulation.

Table 2: Experimental results on Precision with different community datasets for training.(best scores are boldfaced)

Dataset	Android	Academia	Photo	Christian
ASP_SVM	0.7979	0.8054	0.8195	0.7963
ASP_RF	0.8031	0.8265	0.8044	0.8187
ASP_C4.5	0.8002	0.8337	0.8025	0.7846
ASP_Boost	0.7969	0.8271	0.8039	0.8143
ASP_NB	0.7633	0.7835	0.7154	0.7965
MISVM	0.7201	0.7743	0.7982	0.7644
EM-DD	0.7531	0.7557	0.7879	0.7212
BP-MIP	0.7748	0.8153	0.7294	0.7238
MIDL	**0.8113**	**0.8744**	**0.8563**	**0.8195**

Table 3: Experimental results on Recall with different community datasets for training.(best scores are boldfaced)

Dataset	Android	Academia	Photo	Christian
ASP_SVM	0.7544	0.7458	0.8003	0.7914
ASP_RF	0.8612	**0.8361**	0.8014	0.8117
ASP_C4.5	0.7886	0.8053	0.8152	0.8089
ASP_Boost	0.7935	0.7731	0.7841	0.7578
ASP_NB	0.7917	0.7659	0.7452	0.7335
MISVM	0.7382	0.7115	0.8674	0.7361
EM-DD	0.7964	0.7411	0.7238	0.7525
BP-MIP	0.8192	0.7871	0.7493	0.8716
MIDL	**0.9773**	0.7966	**0.8947**	**0.9222**

Table 4: Experimental results on F1-Measure with different community datasets for training.(best scores are boldfaced)

Dataset	Android	Academia	Photo	Christian
ASP_SVM	0.7755	0.7746	0.8098	0.7938
ASP_RF	0.8311	0.8313	0.8029	0.8152
ASP_C4.5	0.7944	0.8193	0.8088	0.7966
ASP_Boost	0.7952	0.7992	0.7939	0.7850
ASP_NB	0.7772	0.7746	0.7300	0.7637
MISVM	0.7290	0.7416	0.8314	0.7500
EM-DD	0.7741	0.7483	0.7545	0.7365
BP-MIP	0.7964	0.8010	0.7393	0.7909
MIDL	**0.8866**	**0.8337**	**0.8751**	**0.8678**

Table 5: Experimental results on Accuracy with different community datasets for training.(best scores are boldfaced)

Dataset	Android	Academia	Photo	Christian
ASP_SVM	0.8091	0.8013	0.7996	0.8146
ASP_RF	0.8253	0.8099	0.8000	0.8501
ASP_C4.5	0.8032	0.7765	0.7842	0.8223
ASP_Boost	0.7953	0.8331	0.8089	0.8347
ASP_NB	0.7554	0.7883	0.7969	0.7839
MISVM	0.7555	0.7828	0.7635	0.7947
EM-DD	0.7742	0.8038	0.7803	0.8092
BP-MIP	0.7798	0.8275	0.7934	0.8251
MIDL	**0.8429**	**0.8563**	**0.8337**	**0.8901**

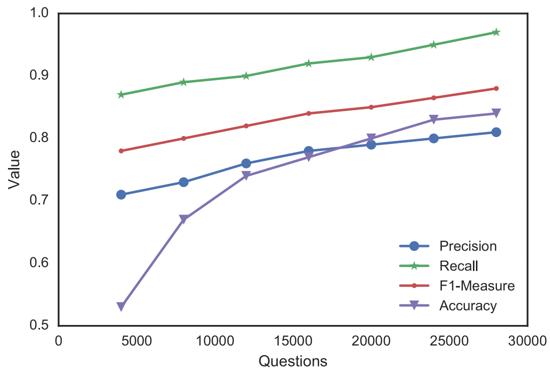

Figure 5: Precision, Recall, F1-Measure, Accuracy for varying amount of training data in Android CQA forum.

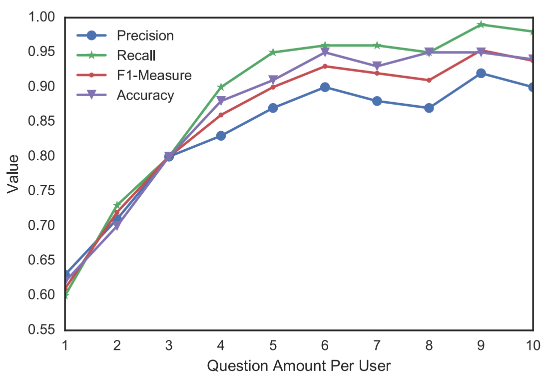

Figure 6: Precision, Recall, F1-Measure, Accuracy for varying active level of users, here we use average questions per user as the group clustering criteria in Android CQA forum.

- Compared with artificial feature selection models, our framework MIDL integrated with deep learning representation gains better experimental results. Moreover, our framework is easier to train with deep learning tactic.
- We implement three typical multiple instance learning algorithms MISVM, EM-DD and BP-MIP. These three algorithms achieved superior performances in their own problem settings. However, in predicting the user's satisfaction towards to bags of answers, they do not work well. We conjecture that this is due to the problem settings and obviously our framework is more appropriate for the user satisfaction prediction task.
- It is no surprising to see from Figure 5 that with sufficient training data, we can achieve a better performance since deep learning method can learn more accurate representations from the big data.
- The accuracy increased with more records for individuals. From Figure 6 we notice that the prediction dramatically increases for users with varying amount of questions. The tendency of the folding lines arise as the number of questions per user increases. And we can clearly see that the folding lines slow down and tend to consistant after reaching 5 quesitons per user. So we can conclude that if we want to obtain a better prediction results, we need at least 5 records for per user.

Overall, compared with other strong baselines, our framework improvements of efficiency give the credit to three aspects. First off, we replace tedious feature selection with deep learning. It enables the extension of databases even without external textual resources. Moreover, the expressive of neural tensor network extracts abundant latent relationships between user oriented question and corresponding answer sets. Our framework can deal with more complicated interactions with tensor layers than the other methods. Last but not least, the assumption of our multiple instance learning framework designed for user satisfaction prediction task is more appropriate in real scenarios.

5. CONCLUSION

Users satisfaction prediction is an essential component in Community Question Answering(CQA) services. Existing approaches have been hurt from the necessaries of predefining artificial selected features, which are usually difficult to design and labor-intensive in real applications. In this paper we formulate the user satisfaction prediction problem as a multiple instance learning pattern, and discuss a new framework which is capable of exploiting deep learning representations associated with our assumption to enhance the weakly supervised learning ability. We develop a neural tensor network based method with Bi-directional LSTM for evaluating the user's attitude towards a set of answers related to the proposed question. Our approach can be applied easily to existing information retrieval models and extended into other user satisfaction modeling field. Experimental results conducted on a large CQA data set from Stack Exchange demonstrate the significant improvement of the proposed technique.

This work opens to several interesting directions for future work. First, it is of relevance to apply the proposed technique to other information retrieval approaches. We notice that in web search engine, recommendation system and social network user behavior analysis all follow the assumption of multiple instance learning. What's more, we can use more complex means to model the users latent preference and enhance the performance. Moreover, applying multiple instance learning with deep learning tactic into Natural Language Processing field is a big treasure to hunt. As future work, we will extend the multiple instance learning assumption into more applicable scenarios.

6. ACKNOWLEDGEMENT

This work is supported by the National Basic Research Program of China (973 Program) under Grant 2013CB336500, National Natural Science Foundation of China under Grant 61602405, 61379071, Fundamental Research Funds for the Central Universities 2016QNA5015 and the China Knowledge Centre for Engineering Sciences and Technology. The Project is also Supported by the Key Laboratory of Advanced Information Science and Network Technology of Beijing (XDXX1603).

7. REFERENCES

[1] J. Andreas, M. Rohrbach, T. Darrell, and K. Dan. Learning to compose neural networks for question answering. 2016.

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

[3] D. Chen, R. Socher, C. D. Manning, and A. Y. Ng. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *arXiv preprint arXiv:1301.3618*, 2013.

[4] T. G. Dietterich, R. H. Lathrop, and T. Lozano-PÃľrez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

[5] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7):2121–2159, 2011.

[6] H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, and M. Ester. Community-based question answering via heterogeneous social network learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[7] A. Hassan, Y. Song, and L.-w. He. A task level metric for measuring web search satisfaction and its application on improving relevance estimation. In *Proceedings of the 20th ACM international conference*, pages 125–134. ACM, 2011.

[8] T. K. Ho. Random decision forests. In *International Conference on Document Analysis and Recognition*, pages 278–282 vol.1, 1995.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] M. Kearns and L. G. Valiant. Crytographic limitations on learning boolean formulae and finite automata. In *ACM Symposium on Theory of Computing*, pages 29–49, 1989.

[11] O. Z. Kraus, J. L. Ba, and B. J. Frey. Classifying and segmenting microscopy images with deep multiple instance learning. In *Bioinformatics*, 2016.

[12] K. Latha and R. Rajaram. Improvisation of seeker satisfaction in yahoo! community question answering portal. *Ictact Journal on Soft Computing*, 1(3), 2011.

[13] L. T. Le, C. Shah, and E. Choi. Evaluating the quality of educational answers in community question-answering. In *The Acm/ieee-Cs*, pages 129–138, 2016.

[14] Q. Liu, E. Agichtein, G. Dror, E. Gabrilovich, Y. Maarek, D. Pelleg, and I. Szpektor. Predicting web searcher satisfaction with existing community-based answers. In *International ACM SIGIR Conference*, pages 415–424, 2011.

[15] Y. Liu, J. Bian, and E. Agichtein. Predicting information seeker satisfaction in community question answering. *Acm Transactions on Knowledge Discovery from Data*, 3(2):pÃągs. 47–52, 2009.

[16] O. Melamud, J. Goldberger, and I. Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *CoNLL*, 2016.

[17] X. Qiu and X. Huang. Convolutional neural tensor network architecture for community-based question answering. In *International Conference on Artificial Intelligence*, 2015.

[18] J. R. Quinlan. C4.5: programs for machine learning. 1993.

[19] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

[20] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. 2013.

[21] A. Vezhnevets and J. M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *IEEE Computer Society Conference on CVPR*, pages 3249–3256, 2010.

[22] B. Wang, M. Ester, J. Bu, Y. Zhu, Z. Guan, and D. Cai. Which to view: Personalized prioritization for broadcast emails. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1181–1190, 2016.

[23] B. Wang, C. Wang, J. Bu, C. Chen, W. V. Zhang, D. Cai, and X. He. Whom to mention: expand the diffusion of tweets by @ recommendation on micro-blogging systems. In *22nd International World Wide Web Conference*, pages 1331–1340, 2013.

[24] H. Wang, Y. Song, M.-W. Chang, X. He, A. Hassan, and R. W. White. Modeling action-level satisfaction for search task satisfaction prediction. In *Proceedings of the 37th international ACM SIGIR conference*, pages 123–132. ACM, 2014.

[25] J. Wu, Y. Yu, C. Huang, and K. Yu. Deep multiple instance learning for image classification and auto-annotation. In *CVPR*, 2015.

[26] L. X. J. Xu and Y. L. J. G. X. Cheng. Modeling document novelty with neural tensor network for search result diversification.

[27] Q. Zhang and S. A. Goldman. Em-dd: An improved multiple-instance learning technique. In *NIPS*, 2001.

[28] Z. Zhao, H. Lu, D. Cai, X. He, and Y. Zhuang. User preference learning for online social recommendation. *IEEE Trans. Knowl. Data Eng.*, 28(9):2522–2534, 2016.

[29] Z. Zhao, Q. Yang, D. Cai, X. He, and Y. Zhuang. Expert finding for community-based question answering via ranking metric network learning. In *IJCAI*, 2016.

[30] Z. Zhao, L. Zhang, X. He, and W. Ng. Expert finding for question answering via graph regularized matrix completion. *IEEE Trans. Knowl. Data Eng.*, 27:993–1004, 2015.

[31] Z. H. Zhou, K. Jiang, and M. Li. Multi-instance learning based web mining. *Applied Intelligence*, 22(2):135–147, 2004.

[32] Z.-H. Zhou and M.-L. Zhang. Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology*, pages 455–459, 2002.