Multimodal DBN for Predicting High-Quality Answers in cQA portals

Haifeng Hu, Bingquan Liu, Baoxun Wang, Ming Liu, Xiaolong Wang
School of Computer Science and Technology
Harbin Institute of Technology, China
{hfhu, liubq, bxwang, mliu, wangxl}@insun.hit.edu.cn

Abstract

In this paper, we address the problem for predicting cQA answer quality as a classification task. We propose a multimodal deep belief nets based approach that operates in two stages: First, the joint representation is learned by taking both textual and non-textual features into a deep learning network. Then, the joint representation learned by the network is used as input features for a linear classifier. Extensive experimental results conducted on two cQA datasets demonstrate the effectiveness of our proposed approach.

1 Introduction

Predicting the quality of answers in community based Question Answering (cQA) portals is a challenging task. One straightforward approach is to use textual features as a text classification task (Agichtein et al., 2008). However, due to the word over-sparsity and inherent noise of user-generated content, the classical bag-of-words representation, is not appropriate to estimate the quality of short texts (Huang et al., 2011). Another typical approach is to leverage non-textual features to automatically identify high quality answers (Jeon et al., 2006; Zhou et al., 2012). However, in this way, the mining of meaningful textual features usually tends to be ignored.

Intuitively, combining both textual and non-textual information extracted from answers is helpful to improve the performance for predicting the answer quality. However, textual and non-textual features usually have different kinds of representations and the correlations between them are highly non-linear. Previous study (Ngiam et al., 2011) has shown that it is hard for a shallow model to discover the correlations over multiple sources.

To this end, a deep learning approach, called multimodal deep belief nets (mDBN), is introduced to address the above problems to predict the answer quality. The approach includes two stages: feature learning and supervised training. In the former stage, a specially designed deep network is given to learn the unified representation using both textual and non-textual information. In the latter stage, the outputs of the network are then used as inputs for a linear classifier to make prediction.

The rest of this paper is organized as follows: The related work is surveyed in Section 2. Then, the proposed approach and experimental results are presented in Section 3 and Section 4 respectively. Finally, conclusions and future directions are drawn in Section 5.

2 Related Work

The typical way to predict the answer quality is exploring various features and employing machine learning methods. For example, Jeon et al. (2006) have proposed a framework to predict the quality of answers by incorporating non-textual features into a maximum entropy model. Subsequently, Agichtein et al. (2008) and Bian et al. (2009) both leverage a larger range of features to find high quality answers. The deep research on evaluating answer quality has been taken by Shah and Pomerantz (2010) using the logistic regression model. We borrow some of their ideas in this paper.

In deep learning field, extensive studies have been done by Hinton and his co-workers (Hinton et al., 2006; Hinton and Salakhutdinov, 2006; Salakhutdinov and Hinton, 2009), who initially propose the deep belief nets (DBN). Wang et.al (2010; 2011) firstly apply the DBNs to model semantic relevance for qa pairs in social communities. Meanwhile, the feature learning for disparate sources has also been the hot research topic. Lee et al. (2009) demonstrate that the hidden representations computed by a convolutional DBN make excellent features for visual recognition.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 843–847,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

3 Approach

We consider the problem of high-quality answer prediction as a classification task. Figure 1 summarizes the framework of our proposed approach. First, textual features and non-textual features extracted from cQA portals are used to train two DBN models to learn the high-level representations independently for answers. The two high-level representations learned by the deep architectures are then joined together to train a RBM model. Finally, a linear classifier is trained with the final shared representation as input to make prediction.

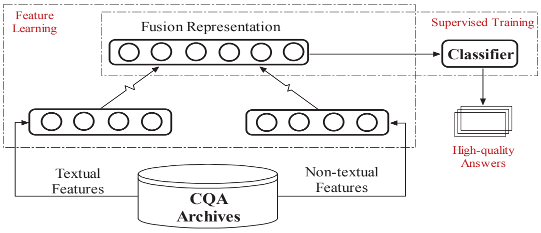

Figure 1: Framework of our proposed approach.

In this section, a deep network for the cQA answer quality prediction is presented. Textual and non-textual features are typically characterized by distinct statistical properties and the correlations between them are highly non-linear. It is very difficult for a shallow model to discover these correlations and form an informative unified representation. Our motivation of proposing the mDBN is to tackle these problems using an unified representation to enhance the classification performance.

3.1 The Restricted Boltzmann Machines

The basic building block of our feature leaning component is the Restricted Boltzmann Machine (RBM). The classical RBM is a two-layer undirected graphical model with stochastic visible units $\mathbf{v}$ and stochastic hidden units $\mathbf{h}$.The visible layer and the hidden layer are fully connected to the units in the other layer by a symmetric matrix $\mathbf{w}$. The classical RBM has been used effectively in modeling distributions over binary-value data. As for real-value inputs, the gaussian RBM (Bengio et al., 2007) can be employed. Different from the former, the hypothesis for the visible unit in the gaussian RBM is the normal distribution.

3.2 Feature Learning

The illustration of the feature learning model is given by Figure 2. Basically, the model consists of two parts.

In the bottom part (i.e., V-H_1, H_1-H_2), each data modality is modeled by a two-layer DBN separately. For clarity, we take the textual modality as an example to illustrate the construction of the mDBN in this part. Given a textual input vector $\mathbf{v}$, the visible layer generates the hidden vector $\mathbf{h}$, by

$$p(h_j = 1|\mathbf{v}) = \sigma(c_j + \sum_i w_{ij} v_i).$$

Then the conditional distribution of $\mathbf{v}$ given $\mathbf{h}$, is

$$p(v_i = 1|\mathbf{h}) = \sigma(b_i + \sum_j w_{ij} h_j).$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ denotes the logistic function. The parameters are updated by performing gradient ascent using Contrastive Divergence (CD) algorithm (Hinton, 2002).

After learning the RBMs in the bottom layer, we treat the activation probabilities of its hidden units driven by the inputs, as the training data for training a new layer. The construction procedures for the non-textual modality are similar to the textual one, except that we use the gaussian RBM to model the real-value inputs in the bottom layer.

Finally, we combine the two models by adding an additional layer, H_3, on the top of them to form the mDBN. The training method is also similar to the bottom's, but the input vector is the concatenation of the mapped textual vector and the mapped non-textual vector.

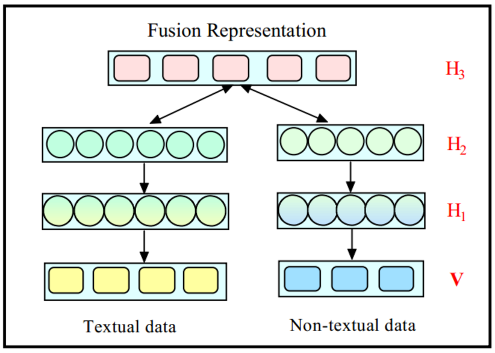

Figure 2: mDBN for Feature Learning

It should be noted in the network, the bottom part is essential to form the joint representation because the correlations between the textual and non-textual features are highly non-linear. It is hard for a RBM directly combining the two disparate sources to learn their correlations.

3.3 Supervised Training and Classification

After the above steps, a deep network for feature learning between textual and non-textual data is established. Classifiers, either support vector machine (SVM) or logistic regression (LR), can then be trained with the unified representation (Ngiam et al., 2011; Srivastava and Salakhutdinov, 2012). Specifically, the LR classifier is used to make the final prediction in our experiments since it keeps to deliver the best performance.

3.4 Basic Features

`Textual Features`: The textual features extract from 1,500 most frequent words in the training dataset after standard preprocessing steps, namely word segmentation, stopwords removal and stemming[1]. As a result, each answer is represented as a vector containing 1,500 distinct terms weighted by binary scheme.

`Non-textual Features`: Referring to the previous work (Jeon et al., 2006; Shah and Pomerantz, 2010), we adopt some features used in theirs and also explore three additional features marked by ‡ sign. The complete list is described in Table 1.

Features	**Type**
Length of question title (description)	Integer
Length of answer	Integer
Number of unique words for the answer ‡	Integer
Ratio of the qa length ‡	Float
Answer's relative position ‡	Integer
Number of answers for the question	Integer
Number of comments for the question	Integer
Number of questions asked by asker (answerer)	Integer
Number of questions resolved by asker (answerer)	Integer
Asker's (Answerer's) total points	Integer
Asker's (Answerer's) level	Integer
Asker's (Answerer's) total stars	Integer
Asker's (Answerer's) best answer ratio	Float

Table 1: Summary of non-textual features.

4 Experiments

4.1 Experiment Setup

Datasets: We carry out experiments on two datasets. One dataset comes from Baidu Zhidao[2], which contains 33,740 resolved questions crawled by us from the "travel" category. The other dataset is built by Chen and Nayak (2008) from Yahoo! Answers[3]. We refer to these two datasets as `ZHIDAO` and `YAHOO` respectively and randomly sample 10,000 questions from each to form our experimental datasets. According to the user name, we have crawled all the user profile web pages for non-textual feature collection. To alleviate unnecessary noise, we only select those questions with number of answers no less than 3 (one best answer among them), and those answers at least have 10 tokens. The statistics on the datasets used for experiments are summarized in Table 2.

Statistics Items	`YAHOO`	`ZHIDAO`
# of questions	6841	5368
# of answers	74485	22435
# of answers per question	10.9	4.1
# of users	28812	12734

Table 2: Statistics on experimental datasets.

Baselines and Evaluation Metrics: We compare against the following methods as our baselines. (1) Logistic Regression (LR): We implement the approach used by Shah and Pomerantz (2010) with textual features `LR-T`, non-textual features `LR-N` and their simple combination `LR-C`. (2) DBN: Similar to the mDBN, the outputs of the last hidden layer by the DBN are used as inputs for LR model. Based on the feature sets, we have `DBN-T` for textual features and `DBN-N` for non-textual features.

Since we mainly focus on the high quality answers, the *precision*, *recall* and *f1* for positive class and the overall *accuracy* for both classes are employed as our evaluation metrics.

Model Architecture and Training Details: To create the mDBN architecture, we use the classical RBM with 1500 visible units followed by 2 hidden layers with 1000 and 800 units respectively for the textual branch, and the gaussian RBM with 20 visible units followed by 2 hidden layers with 100 and 200 units respectively for the non-textual branch. On the joint layer of the network, the layer contains 1000 real-value units.

Each RBM is trained using 1-step CD algorithm. During the training stage, a small weight-cost of 0.0002 is used, and the learning rate for textual data modal is 0.05 while the non-textual data is 0.001. We also adopt a monument of 0.5 for the first five epochs and 0.9 for the rest epochs. In addition, all non-textual data vectors are normalized to have zero mean and unit standard variance. More details for training the deep architecture can be found in Hinton (2012).

4.2 Results and Analysis

In the first experiment, we compare the performance of mDBN with different methods. To make a fare comparison, we use the liblinear toolkit[4] for logistic regression model with L2 regularization and randomly select 70% QA pairs as training data

[1] The stemming step is only used in English corpus.

[2] http://zhidao.baidu.com

[3] http://answers.yahoo.com

[4] available at http://www.csie.ntu.edu.tw/ cjlin/liblinear

and the rest 30% as testing data. Table 3 and Table 4 summarize the average results of the 5 round experiments on YAHOO and ZHIDAO respectively.

Methods	P	R	F1	Accu.
LR-T	0.374	0.558	0.448	0.542
LR-N	0.524	0.614	0.566	0.686
LR-C	0.493	0.557	0.523	0.662
DBN-T	0.496	0.571	0.531	0.663
DBN-N	0.505	0.578	0.539	0.670
mDBN	**0.534**	**0.631**	**0.579**	**0.694**

Table 3: Comparing results on YAHOO

It is promising to see that the proposed mDBN method notably outperforms almost all the other methods on both datasets over all the metrics as expected, except for the *recall* on ZHIDAO. The main reason for the improvements is that the joint representation learned by mDBN is able to complement each modality perfectly. In addition, the mDBN can extract stronger representation through modeling semantic relationship between textual and non-textual information, which can effectively help distinguish more complicated answers from high quality to low quality.

Methods	P	R	F1	Accu.
LR-T	0.380	0.540	0.446	0.553
LR-N	0.523	0.735	0.611	0.688
LR-C	0.537	0.695	0.606	0.698
DBN-T	0.527	0.730	0.612	0.692
DBN-N	0.539	**0.760**	0.631	0.703
mDBN	**0.590**	0.755	**0.662**	**0.743**

Table 4: Comparing results on ZHIDAO

The classification performance of the textual features are worse on average compared with non-textual features, even when the feature learning strategy is employed. More interestingly, we find the simple combinations of textual and non-textual features don't improve the classification results compared with using non-textual features alone.We conjecture that there are mainly three reasons for the phenomena: First, this is due to the fact that user-generated content is inherently noisy with low word frequency, resulting in the sparsity of employing textual feature. Second, non-textual features (e.g., answer length) usually own strongly statistical properties and feature sparsity problem can be better relieved to some extent. Finally, since correlations between the textual features and non-textual features are highly non-linear, concatenating these features simply sometimes can submerge classification performance. In contrast, mDBN enjoys the advantage of the shared representation between textual features and non-textual features using the deep learning architecture.

We also note that neither the mDBN nor other approaches perform very well in predicting answer quality across the two datasets. The best *precision* on ZHIDAO and YAHOO are respectively 59.0% and 53.4%, which means that there are nearly half of the high quality answers not effectively identified. One of the possible reason is that the quality of the corpora influences the result significantly. As shown in Table 2, each question on average receives more than 4 answers on ZHIDAO and more than 10 on YAHOO. Therefore, it is possible that there are several answers with high quality to the same question. Selecting only one as the high quality answer is relatively difficult for our humans, not to mention for the models.

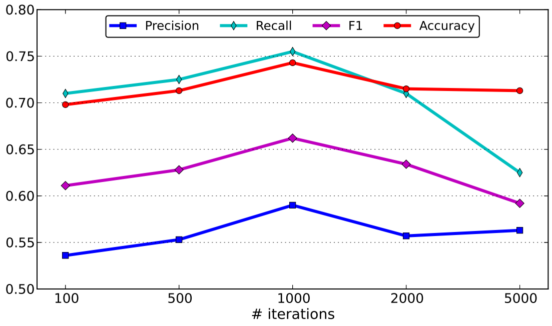

Figure 3: Influences of iterations for mDBN

In the second experiment, we intend to examine the performance of mDBN with different number of iterations. Figure 3 depicts the metrics on ZHIDAO when the iteration number is varied from 100 to 5000. From the result, the first observation is that increasing the number of iterations the performance of mDBN can improve significantly, obtaining the best results for iteration of 1000. This clearly shows the representation power of the mDBN again. However, after a large number of iterations (large than 1000), the mDBN has a detrimental performance. This may be explained by with large number of iterations, the deep learning architecture is easier to be overfitting. The similar trend is also observed on YAHOO.

5 Conclusions and Future work

In this paper, we have provided a new perspective to predict the cQA answer quality: learning an informative unified representation between textual and non-textual features instead of concatenating them simply. Specifically, we have proposed a multimodal deep learning framework to form the unified representation. We compare this with the basic features both in isolation and in combination. Experimental results have demonstrated that our proposed approach can capture the complementarity between textual and non-textual features, which is helpful to improve the performance for cQA answer quality prediction.

For the future work, we plan to explore more semantic analysis to approach the issue for short text quality evaluation. Additionally, more research will be taken to put forward other approaches for learning multimodal representation.

Acknowledgments

The authors are grateful to the anonymous reviewers for their constructive comments. Special thanks to Chengjie Sun and Deyuan Zhang for insightful suggestions. This work is supported by National Natural Science Foundation of China (NSFC) via grant 61272383 and 61100094.

References

E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. 2008. Finding high-quality content in social media. In *Proceedings of the international conference on Web search and web data mining*, pages 183–194. ACM.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153–160.

Jiang Bian, Yandong Liu, Ding Zhou, Eugene Agichtein, and Hongyuan Zha. 2009. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th international conference on World wide web*, pages 51–60. ACM.

L. Chen and R. Nayak. 2008. Expertise analysis in a question answer portal for author ranking. In *International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 134–140.

G.E. Hinton and R.R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

G.E. Hinton, S. Osindero, and Y.W. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

G.E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

G.E. Hinton. 2012. A practical guide to training restricted boltzmann machines. *Lecture Notes in Computer Science*, pages 599–619.

Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2011. Quality-biased ranking of short texts in microblogging services. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 373–382.

J. Jeon, W.B. Croft, J.H. Lee, and S. Park. 2006. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 228–235. ACM.

H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A.Y. Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 689–696.

R. Salakhutdinov and G.E. Hinton. 2009. Deep boltzmann machines. In *Proceedings of the international conference on artificial intelligence and statistics*, volume 5, pages 448–455.

C. Shah and J. Pomerantz. 2010. Evaluating and predicting answer quality in community qa. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 411–418.

N. Srivastava and R. Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2231–2239.

B. Wang, X. Wang, C. Sun, B. Liu, and L. Sun. 2010. Modeling semantic relevance for question-answer pairs in web social communities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1230–1238. ACL.

B. Wang, B. Liu, X. Wang, C. Sun, and D. Zhang. 2011. Deep learning approaches to semantic relevance modeling for chinese question-answer pairs. *ACM Transactions on Asian Language Information Processing*, 10(4):21:1–21:16.

Z.M. Zhou, M. Lan, Z.Y. Niu, and Y. Lu. 2012. Exploiting user profile information for answer ranking in cqa. In *Proceedings of the 21st international conference on World Wide Web*, pages 767–774. ACM.