Generating Questions from Web Community Contents

Baoxun Wang[1] *Bingquan Liu*[1]
Chengjie Sun[1] *Xiaolong Wang*[1] *Deyuan Zhang*[2]
(1) School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
(2) School of Computer, Shenyang Aerospace University, Shenyang, China
{bxwang, liubq, cjsun, wangxl, dyzhang}@insun.hit.edu.cn

ABSTRACT
Large amounts of knowledge exist in the user-generated contents of web communities. Generating questions from such community contents to form the question-answer pairs is an effective way to collect and manage the knowledge in the web. The parser or rule based question generation (QG) methods have been widely studied and applied. Statistical QG aims to provide a strategy to handle the rapidly growing web data by alleviating the manual work. This paper proposes a deep belief network (DBN) based approach to address the statistical QG problem. This problem is considered as a three-step task: *question type determination*, *concept selection* and *question construction*. The DBNs are introduced to generate the essential words for question type determination and concept selection. Finally, a simple rule based method is used to construct questions with the generated words. The experimental results show that our approach is promising for the web community oriented question generation.

KEYWORDS: statistical question generation, deep belief network, web community.

Proceedings of COLING 2012: Demonstration Papers, pages 467–474,
COLING 2012, Mumbai, December 2012.

1 Introduction

Automatic question generation (QG) is a challenging task in the NLP field, and its difficulties are being realized by the researchers gradually. Since 2008, the workshop on QG[1] has been offering the shared task and evaluation on this problem. At present, the QG technique tends to be mainly applied in the interaction oriented systems (Rus et al., 2007; Harabagiu et al., 2005) (e.g., computer aided education, help desk, dialog systems, etc.). In most systems, the original source texts are parsed and transformed into the questions with the rules. The parser and rule based methods always maintain considerable accuracy, and the generating results can be directly presented to the users.

In this paper, we aim to address the web-community oriented question generation in a statistical learning way. A deep belief network (DBN) is proposed to generate the essential elements of the questions according to the answers, based on the joint distributions of the questions and their answers learned by the network from a number of user-generated QA pairs in the web communities. The generated words are then reorganized to form the questions following some simple rules.

The rest of this paper is organized as follows: Section 2 surveys the related work. Section 3 details our approach to question generation. Experimental results are given and discussed in Section 4. Finally, conclusions and future directions are drawn.

2 Related Work

To our knowledge, there is no previous work that concentrates on statistically generating questions from the web content freely posted by users, as we do in this paper. Nevertheless, the basic work has been done on the definition and evaluation of automatic QG. Nielsen (2008) gives the definition of QG and considers this problem as a three-step process. A question taxonomy for QG is proposed in (Nielsen et al., 2008) with a detailed question branch offered. The overall description of the QG task is proposed in (Rus and Graesser, 2009; Rus et al., 2010). Rus et al. (2007) and Vanderwende (2008) have discussed the evaluation of the QG systems.

The technique of question generation is essential to some education related fields, such as educational assessment, intelligent tutoring, etc. Brown et al. (2005) have described an approach to automatically generating questions for vocabulary assessment. Hoshino and Nakagawa (2005) have developed a real-time system which generates questions on English grammar and vocabulary. A template based QG method is proposed in (Wang et al., 2008) to evaluate the learners' understanding after reading a medical material. In conclusion, the purpose of such QG systems is different from our goal in this paper. It should be noted that the nature of automatic question generation is different depending on the application within which it is embedded (Nielsen, 2008).

3 Generating Questions using the Deep Belief Network

Nielsen (2008) defines the question generation task as a 3-step process: *question type determination*, *concept selection* and *question construction*. Basically, the architecture of our work follows this definition. Figure 1 illustrates the framework of our QG research: the human-generated QA pairs crawled from the cQA portals are used to train two DBN models to generate the question words (5W1H[2]) and the content words independently for the input source

[1] http://www.questiongeneration.org

[2] 5W1H stands for the 6 common question words in English: what, when, where, who, why, and how.

knowledge text. The essential words automatically generated by the deep networks are then organized with the manually written patterns to form the final questions.

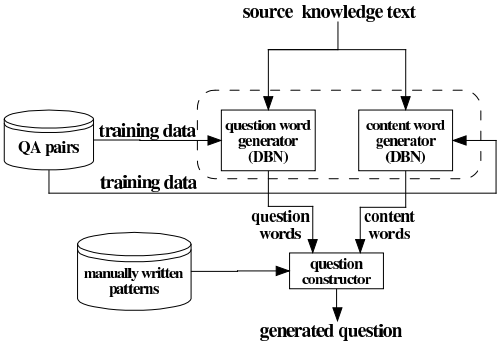

Figure 1: Framework of our statistical QG approach.

In this section, a deep network for the community content oriented QG is presented. Given the word occurrence information of an answer, the trained networks are expected to generate the words of the question. Our motivation of proposing the DBN to handle the QG problem is to build the semantic links between the questions and the answers. Intuitively, the words in the answers are helpful to provide the clues for predicting the essential words of the corresponding questions. Our DBN models are designed to learn the semantic relationships of the words in the QA pairs, and obtain the hidden clues in a statistical way. In detail, we utilize the ability of the deep network to map the QA pairs into a semantic feature space, where the joint distributions of the questions and their answers are modeled.

3.1 The Restricted Boltzmann Machine

A DBN is composed of several stacked "Restricted Boltzmann Machines". The Restricted Boltzmann Machine (RBM) can be used to model an ensemble of binary vectors (Hinton, 2002; Hinton and Salakhutdinov, 2006). Salakhutdinov and Hinton (2009) have proposed a deep graphical model composed of RBMs into the information retrieval field, which shows that this model is able to obtain semantic information hidden in the word-count vectors. The RBM is a two-layer network(Hinton, 2002), the bottom layer represents a visible vector $\mathbf{v}$ and the top layer represents a latent feature vector $\mathbf{h}$. The matrix W contains the symmetric interaction terms between the visible units and the hidden units. In the RBM, the visible feature vectors can be used to compute the "hidden features" in the hidden units. The RBM model can reconstruct the inputs using the hidden features.

3.2 Training a Deep Belief Network for QG

Our work is inspired by (Hinton et al., 2006), which proposes a DBN for image labeling. In their research, the trained deep net is able to give the labels based on the input images. The illustration of our DBN model is given by Figure 2. Basically, this model is composed of three

layers, and here each layer stands for the RBM described in Subsection 3.1. In this network, the function of the bottom layer and the middle layer is to reduce the dimension of the visible answer vectors by mapping them into a low-dimensional semantic space. The top layer is essential to the QG task, since the question vectors and the mapped answer vectors are joined together and the joint distribution of QA pairs are modeled in this layer.

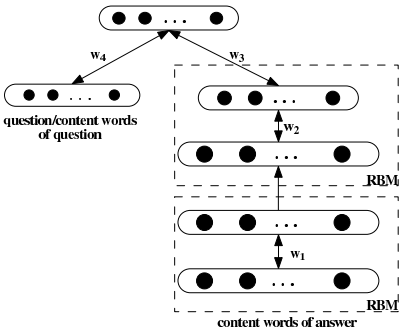

Figure 2: The DBN for Question Generation.

Here we take the bottom layer as an example to explain the pretraining procedure of the DBN, because the computing procedures of the three layers are indeed the same. Given the training set of binary answer vectors based on the statistics of the word occurrence, the bottom layer generates the corresponding hidden features. The hidden features are then used to reconstruct the Bernoulli rates for each word in the answer vectors after stochastically activating the hidden features. Then the hidden features are activated with the reconstructed input vectors. We use 1-step Contrastive Divergence (Hinton, 2002) to update the parameters by performing gradient ascent. After training one layer, the **h** vectors are then sent to the higher-level layer as its "training data". The training method of the rest two layers is the same with the bottom's. It should be noted is that in the top layer, the question vector and the mapped answer vector are joined together to form a new vector as the "input vector", and their weight matrixes are concatenated correspondingly.

During the pre-training procedure, a greedy strategy is taken to train each layer individually, so it is necessary to fine-tune the weights of the entire network. In order to tune the weights, the network is unrolled, taking the answers as the input data to generate the corresponding questions at the output units. Using the cross-entropy error function, the network can be tuned by performing back propagation through it and the objective is to reduce the generation error further. The procedure is the same with that described in (Hinton et al., 2006).

3.3 Generating the Questions

The word predicting work can be completed with the trained deep networks: we send the binary answer vectors to the right branch of the DBN to perform a level-by-level computation. In the top layer, the network gets the top feature vector using the mapped answer vector. With the top feature, the model performs a reverse computation to generate the real-value vector

Question Word	Pattern
how	how+to+< $verb$ >+< adj^* >+< $noun$ >+?
what	what+to+< $verb$ >+for+< adj^* >+< $noun$ >+? what+is+< adj^* >+< $noun$ >+to+< $verb$ >+< $noun$ >+?
where	where+can+i+< $verb$ >+< adj^* >+< $noun$ >+?
how much	how much+for+< $verb$ >+< adj^* >+< $noun$ >+?
how many	how many +< adj^* >+< $noun$ >+to+< $verb$ >+?

Table 1: Examples of the patterns for question construction.

at the output question units. To get the question word, we only have to find the one with the highest value. In order to obtain the content words, the generated values need to be sorted in descending order, then the top ranked *n* words are selected.

A collection of patterns guided by the question words is used to accomplish the final step of QG. The question words influence not only the order of the generated content words, but also the selection and the positions of the additional function words (e.g., prepositions). Table 1 gives some examples of the patterns designed by us. The spirit of the patterns is to reorganize the content words generated by the DBN, and help to add necessary function words, based on the guiding question words.

4 Experiments

4.1 Experimental Setup

Dataset: In this paper, the datasets come from the "Travel" category of Yahoo! Answers. We have chosen 4 different topics: *trip plans*, *cutting down expense*, *necessary items*, and *VISA application*. Based on each topic, various queries are submitted to the cQA service and the "resolved questions" with their best answers are crawled. After filtering the questions whose answers are less than 10 words or containing the URLs only, from each topic we get 4,500 QA pairs for training and 100 randomly selected QA pairs for testing.

Baseline: Noticing that there are no previous studies focusing on the statistical QG, this paper introduces three popular statistical methods as the baselines to predict the essential words of questions for comparison: *Naive Bayesian*, *K-Nearest Neighbor*, and *Latent Semantic Indexing*.

Evaluation: The performance of the network generating the question words is evaluated by calculating the precision of the generated results; and the performance of the content word generation is evaluated by calculating the ratio of the number of the successful generations to the number of the total generations strictly. In this procedure, only if all the top 3 generated content words appear in the original question sentence, the generation is considered to be successful, otherwise, the generation is considered to be unsuccessful.

4.2 Results and Analysis

Table 2 lists the evaluating results for the question word generation (QWG) task and the content word generation (CWG) task on our datasets from Yahoo! Answers. From the tables, it can be observed that our DBN based model has outperformed the baseline methods as expected, which shows the considerable potential of our approach on the web content oriented question generation. From the user-generated QA pairs, the networks eventually learn the semantic knowledge for modeling the joint distributions of the questions and their answers. Due to the effective modeling work, the DBNs can predict the words in a question when given the corre-

Method	Precision of essential word generation							
	trip plans		cutting down expense		necessary items		VISA application	
	QWG	CWG	QWG	CWG	QWG	CWG	QWG	CWG
NB	0.19	0.28	0.27	0.38	0.23	0.32	0.28	0.35
KNN	0.22	0.26	0.36	0.45	0.25	0.36	0.33	0.42
LSI	0.25	0.30	0.42	0.48	0.32	0.41	0.37	0.46
DBN	0.36	0.51	0.62	0.72	0.57	0.69	0.53	0.58

Table 2: Results of question / content word generation on the datasets from Yahoo! Answers.

sponding answer, although the lexical gaps exist between them and the feature sparsity is a common problem.

We can see that our method's precision of content word generation is higher than that of question word generation. This is reasonable because it tends to be easier to obtain the semantic links between the content words in the questions and those in the answers. For the question words, however, the clues hidden in the answers for prediction are less obvious. In this situation, the average precision of QWG has reached 52%, which indicates the deep network's ability to acquire the hidden semantic features.

original	How do I go about planning my trip to Machu Picchu?
generated	How to plan a trip (travel) to Machu Picchu?
original	What should i pack for travel to Paris, and i'm a woman?
generated	What to take for the travel to Paris as a woman?
original	How much money will I need for the trip to Greece?
generated	How much money for taking the trip to Greece?

Table 3: Question samples generated from the answers in the cQA corpora.

To show the performance of our generating approach directly, some generating samples are given in Table 3. In this table, the questions in the QA pairs from our testing data are taken as the original questions, and the corresponding answers are used to get the generated questions. As shown in Table 3, the generated questions are mostly shorter than the original ones. The reason is that our methodology focuses on the major contents of the object question sentences, and the less important contents are ignored.

Conclusions

In this paper, we have proposed a deep belief network based statistical approach to generating questions according to the user-generated web community contents. The contributions of this paper can be summarized as follows: (1) this paper has presented a statistical method for web community oriented question generation. (2) by modeling the joint distributions of the QA pairs, the deep network is able to learn the semantic relationship between the words in the questions and their answers, so as to generate the essential words of the questions.

Acknowledgement

The authors are grateful to the anonymous reviewers for their constructive comments. Special thanks to Dr. Rodney D. Nielsen. This work is supported by the National Natural Science Foundation of China (61100094 and 61272383), Research Fund for the Doctoral Program of Higher Education of China (20102302120053).

References

Brown, J., Frishkoff, G., and Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In *Proceedings of EMNLP'05: HLT*, pages 819–826, Vancouver, British Columbia, Canada. ACL.

Harabagiu, S., Hickl, A., Lehmann, J., and Moldovan, D. (2005). Experiments with interactive question-answering. In *In Proceedings of ACL 05*, pages 60–69.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14.

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

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

Hoshino, A. and Nakagawa, H. (2005). A real-time multiple-choice question generation for language testing: a preliminary study. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, EdAppsNLP 05, pages 17–20, Morristown, NJ, USA. ACL.

Nielsen, R., Buckingham, J., Knoll, G., Marsh, B., and Palen, L. (2008). A taxonomy of questions for question generation. In *Proceedings of Workshop on the Question Generation Shared Task and Evaluation Challenge*.

Nielsen, R. D. (2008). Question generation: Proposed challenge tasks and their evaluation. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, Virginia.

Rus, V., Cai, Z., and Graesser, A. C. (2007). Evaluation innatural language generation: The question genera-tion task. In *Proceedings of Workshop on Shared Tasks and Com-parative Evaluation in Natural Language Generation*.

Rus, V. and Graesser, A. (2009). The question generation task and evaluation challenge. Technical report, Institute for Intelligent Systems.

Rus, V., Wyse, B., Piwek, P., Lintean, M., Stoyanchev, S., and Moldovan, C. (2010). The first question generation shared task evaluation challenge. In *Proceedings of the Sixth International Natural Language Generation Conference (INLG 2010)*, Trim Castle, Ireland.

Salakhutdinov, R. and Hinton, G. (2009). Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978.

Vanderwende, L. (2008). The importance of beingimportant: Question generation. In *Proceedings of the Workshop on the Question Generation Shared Taskand Evaluation Challenge*.

Wang, W., Hao, T., and Liu, W. (2008). Automatic question generation for learning evaluation in medicine. In *Advances in Web Based Learning,ICWL 2007*, volume 4823 of *Lecture Notes in Computer Science*, pages 242–251. Springer Berlin / Heidelberg.