# How to Improve the Answering Effectiveness in Pay-for-Knowledge Community: An Exploratory Application of Intelligent QA System

Yihang Cheng
Tianjin University
chengyihang@tju.edu.cn

Xi Zhang
Tianjin University
jackyzhang@tju.edu.cn

Hao Wang
360 Search Lab
cashenry@126.com

Shang Jiang
Tianjin University
shanjiangtju@tju.edu.cn

## ABSTRACT

Community Question Answering (CQA) has emerged recently and it becomes popular among people. During the process of the communication, different knowledge can be merged. Recently, several vendors use the business model of paying for knowledge to make these knowledge to the monetary benefits, then the Pay-for-Knowledge Communities (PKC) have been applied. Even PKC has interesting business model, there are several problems to be solved, and one of the most salient problem is that questioners may takes too long time to choose the most valuable answers, leading to questioners not able to pay for suitable answerers. There are several previous research has focused on this problem but still have not found satisfactory solutions as questions and answers become more and more complex in the platforms. With the development of cognitive computing techniques, applying an intelligent QA system in PKC to improve the answering effectiveness may be possible. In this paper, we tried to investigate how to apply the intelligent QA system into PKC platform to improve the answering effectiveness. For solving the problems of matching complex questions and answers, we present a Four Module QA Model based on the normal intelligent QA System. Compared to normal intelligent QA System, our model uses categories to classify the questions with traditional machine learning methods. We use answers in each category of corresponding questions as one dataset, answers in each entity of corresponding question as the other dataset, finally, these two datasets make up the document database. Then we got the best answer among past answers through comparing the TF-IDF weighted bag-of word vectors of two datasetsor the new answer including key words through Long Short-Term Memory (LSTM) algorithm with PKC's features composed of centrality and money. Exploratory experiments were developed on a dataset with 1222 users' QA sites collected from a QA community. The model we proposed is expected to increase QA's effectiveness and improve the business model of Pay-for-Knowledge Communities.

## 1 INTRODUCTION

Recently, Community Question Answering (CQA) has become a trend in knowledge and learning applications [1]. In Yahoo! Answer, more than 80 million questions have been resolved by the community [2]. Similarly, there are Baidu Knows, 360 QA and some other CQA platforms in the world.

There are several reasons for the CQA's success: 1) The exact and personalized answer: If people can get the answer only fits them, they will be happy to use this way again. 2) The social needs: CQA platforms have likes and comment functions, which can provide the chance of developing social connections among users. 3) The knowledge creation: The CQA platform provides a communication environment for talents in different fields to create new interdisciplinary knowledge.


ACM ISBN 978-1-4503-5640-4/18/04.
DOI: https://doi.org/10.1145/3184558.3186342

Considering the success of the early CQA platforms, practitioners then applied the business model of pay for knowledge into platform. There are two ways in the process of paying for knowledge: 1) The first is user paying for experts who may answer users' questions correctly, called the systematic pay for knowledge community (PKC) such as *Fenda* and *Ximalaya*; 2) The second way is user paying for the answerers, whose answer is best whatever who they are, called the reward-function PKC, such as *Zhihu* or *Baidu Knows*, which processes are illustrated at Figure 1. The biggest difference of them is whether users know the answers in advance, the first one is yes, and the second is on the contrary.

Even numerous studies have focused on PKC, a fundamental question has not been well answered: 1) Do users get what they really want? 2) If not, what are reasons and what can we do to improve the users' satisfaction effectively? As for the first question, in the process of the Question Answering (QA) in platform, it is likely that they cannot get answer because of the topic, question length or the degree of clarity of explanation to the question [2]. As for the second question, there are many people asking some fundamental questions [3], such as "Who is the US president?", which are too easy to appeal attention. These problems may cause the low answering ratio of these platforms.

Previous work uses different approaches to solve these problems, but most of work can not satisfy the users as giving correct and personalized answers in time. With the development of the intelligent QA system, the application of machine intelligence in PKC platform may be an effective way. But there are also challenges, the most critical one is how to combine the machine intelligence and PKC platform in a suitable way.

This paper mainly considers the intelligent QA system as the manifestations of machine intelligence, combining the intelligent QA and PKC platforms. Our work mainly uses the category to classify the questions users post and Long Short-Term Memory (LSTM) with PKC platform features to train the answers in each category and each entity of corresponding question based on the fundamental QA System. Then, we can answer the question users propose using the key words our training model gets.

## 2 RELATED WORK

### 2.1 Previous work on Pay-for-Knowledge Communities

The Pay-for-Knowledge Communities (PKC) is a new application. Generally, PKC are developed from normal CQA platform, but are more specific and advanced. As rare studies have focused on this specific platform, we will discuss the literature from CQA.

Within CQA, information seekers can get answers by posting questions to other participants and letting them answer. However, CQA is not always effective: in some cases, the user can get the perfect answer in a few minutes, hours or even days [4]. Prior works mainly aimed to find reasons for no-answered questions. Yang et al [2] found the reasons for no-answered questions are depends on the content features, heuristic features such as length and time. There are also some previous works aiming at matching answerers with questions. Li and King [5] proposed a framework called Question Routing (QR) to route questions to answerers who are top ranking according to their performance in previous days.

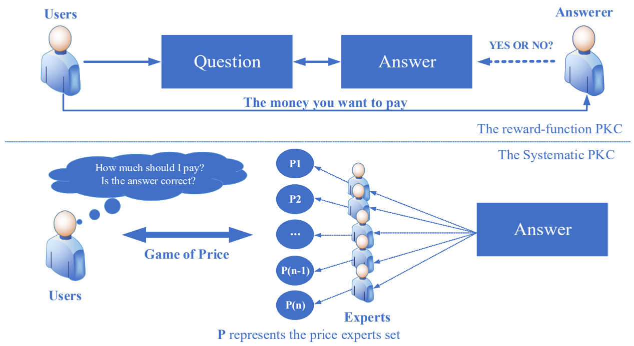


**Figure 1**:**The process of the PKC platform**

As for the problems about finding suitable experts in PKC, Riahi et al [6] used Latent Dirichlet Allocation model and Segmented Topic Model to route questions correctly to experts which showed that statistical topic models help get better expertise recommendation. Zhang [7] modeled the QA community into an expertise graph, and proposed an Expertise Ranking algorithm. Yang et al [8] proposed Topic Expertise Model (TEM) to model topics and expertise together by integrating textual content model and link structure analysis. These ways indeed improve the quality of answers and the answers, but the effectiveness of them decreases fast as the questions and answers are becoming more and more complex.

## 2.2 Intelligent QA System

Previous works mainly focus to improve the answering effectiveness in the existing QA environment. If the questions and answers become more complex, the waiting time may be longer. With the development of the cognitive computing (i.e., a method for computing simulating the mechanisms of the brain [9]), intelligent QA systems were presented, by which intelligently QA can decrease the waiting time and improve the effectiveness. Combination of intelligent QA system and the PKC platform are the implement of cognitive computing. The researches about intelligent QA system are mainly aiming to increase the precision of answering.

Chen et al [10] use the TF-IDF in retrievers and the RNN in readers to successfully make machine answer the open-domain questions in exact-match 69.5% and F1-score 78.8% for the Stanford Question Answering Dataset (SQuAD). These ratios reflect the machine can answer our fundamental questions. To improve the effectiveness of answering questions, Eunsol et al [11] also used the summary of documents in reader so that long documents can be used effectiveness. As for the improvement of the quality of answers, He et al [12] proposed an end-to-end QA system. As for the improvement of questions dealing with, based on the normal QA system, Cui et al [3] propose KBQA model which uses category and template to classify the questions. Hao et al [13] propose a new representation of question combining the knowledge bases (KBs).

Even more and more researchers pay attentions on intelligent QA systems, rare studies focus on integrating the PKC platform and intelligent QA system. In this paper, we tries to apply the intelligent QA system into PKC platform. Our work may have some potential contributions in the field of PKC as follows:

1. We propose the four modules model to describe the intelligent QA system in PKC, which are Question Module, Document Module, Learning Module and Matching Module.Especially, the Learning Module has the features composed of centrality and money in PKC platform and Matching Module is the quite new module in our model compared with normal intelligent QA system.

2.We add PKC's features composed of the number of likes and commentsinto intelligent QA system so that our model can suit the real environment of PKC platform.

# 3 PROBLEM FORMULATION

In this section, we propose our research problem in technical ways. Table 1 displays all definition and notations.

Definition 1: Answer. Answer refers to the process of answering the question. The text or voice answered by experts or someone who knows is called contents represented by $C_a$, the time from posting a question to be answered is represented by $\mathrm{T}_a$, the topic of the correspond question is represented by $\mathrm{TP}_a$, the person proposing this answer is called answerer, which is represented by $\mathrm{A}_a$. Each answer has these four features, so it is displayed as a four tuple $(C_a, T_a, TP_a, A_a)$. All the Answers makes up a dataset called A.

$$\mathrm{A} = \left\{Answer \middle| \left(C_a, T_a, TP_a, A_a\right) C_a, TP_a, A_a \in String T_a \in Int\right\}.$$

Definition 2: Question. Question refers to the process of posting the question, also including many features. The contents are represented by $C_q$, the time that question is posted is represented by $T_q$, the topic of the correspond question is represented by $TP_q$, the person posted this question is called asker, which is represented by $A_q$. The different feature with Answer is the category, which means the specific classification under topic, such as friendship in emotion, is represented by $CG_q$.Similarly, it is displayed as a five tuple $\left(C_q, T_q, TP_q, A_q, CG_q\right)$. All the Questions makes up a dataset called Q,

$$\mathrm{Q} = \left\{Question \middle| \left(C_q, T_q, TP_q, A_q, CG_q\right) C_q, TP_q, A_q, CG_q \in String T_q \in Int\right\}.$$

After describing the definition and notations, we can describe the problem in the process. We formulate the problem in the unified ways as follows.

Problem (Question Answering): When a new Question appears in the PKC platform, we can use the model to provide an Answer, the $C_a$of this Answer is some key words. The results of this problem can be evaluated by the comparison of the existing Answer and the Answer machine provides.

**Table 1: Notations of the definitions**

| DFINITION | SYMBOL | DESCRIPTION |
|---|---|---|
| **Answer** | $C_a$ | Contents of the answer |
| | $\mathrm{T}_a$ | The time that question is answered |
| | $\mathrm{TP}_a$ | The topic of the question answer corresponding |
| | $\mathrm{A}_a$ | The person proposing this answer |
| **Question** | $C_q$ | Contents of the question |
| | $T_q$ | The time that question is posted |
| | $TP_q$ | The topic of the question |
| | $A_q$ | The person posting this question |
| | $CG_q$ | The category under some topic |

# 4 OUR MODEL

## 4.1 Fundamental Framework

Intelligent QA system mainly has two parts: retriever and reader [10]. Based on the fundamental intelligent QA system, we can develop a new and complex model including 4 modules: Question module, Document module, Learning module, and Matching module (Figure 2). Question module is to build the mapping of the category and the template of questions with clustering algorithm and Bayesian probability; document module is to construct training set including answers of different categories and different entities among questions; learning module will learn the dataset's feature of document module by LSTM algorithm; finally, matching module is the process to match suitable words of training results of each category and training results of each entity in question, then, the prediction with key words can be got. The combination of these four modules and matching two results of different training datasets are the biggest contribution of our work. The principle of each module is explained in Section 4.2.

## 4.2 The principle of our model

In this part, we mainly explain the principle of four modules, and construct the complete process of the lifecycle of a new question in our model in the end.

***1. The Question Module:***

Cui et al [3] proposed a KBQA to deal with the question, they use the maximal probability and the category to classify questions. The category and template are given new definitions.

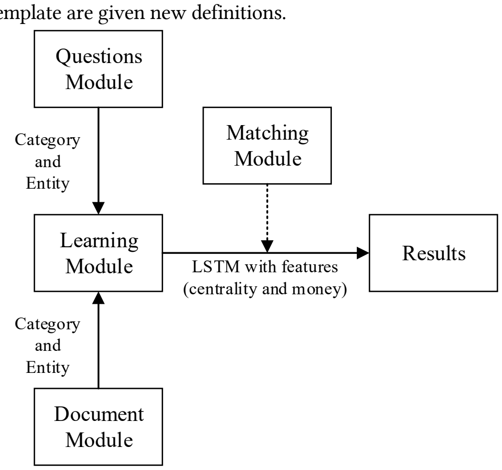


**Figure 2: Overview of our fundamental framework**

In our model, $CG_q$ is the classification for topic, for example, emotion, friendship, love and so on. In the same time, we can extract the entity of every question to get the template of this question. For example, the question "I have a problem about computer, how to open the MAC and how to get the factory code of this computer?" Through the extraction, the templates of this question are "how to open the $entity?" and "how to get $entity?"

Then we can use the K-means clustering to cluster the category, and we can get all categories under the corresponding topic. In the end, we use Bayesian probability to estimate the category of the template, specific formula is as follows.

$$\mathrm{P}\big(y = CG_q|template\big) = \frac{\Pi_{i=1}^{m} P\big(template^i|y = CG_q\big)P\big(y = CG_q\big)}{\Sigma_Q P\big(y = CG_q\big)\,\Pi_{i=1}^{m} P\big(template^i|y = CG_q\big)}$$

$$\mathrm{P} = \max\begin{pmatrix} \mathrm{P}\big(y = CG_{q1}|template\big), \\ \mathrm{P}\big(y = CG_{q2}|template\big)\ldots\mathrm{P}\big(y = CG_{qQ}|template\big) \end{pmatrix}$$

P is the maximum probability, and corresponding category is which the template belongs. In this formula, $P\big(y = CG_q\big)$and $P\big(template^i|y = CG_q\big)$can be learned by the existing dataset. Until now, each question has its own $TP_q$and $CG_q$besides $C_q$.

***2. Document Module and Learning Module:***

Because the Learning Module is to learn the results of Document Module, explaining these two modules together can be explained logically and understood easily. We add PKC's features composed of centrality and money into LSTM algorithm and take two trainings according to different categories and different entities.

As for the questions in PKC, there are usually two kinds: one is that question has already answered in past, the other is that question has not answered which means it's new.

**(1) The question has already answered:**

The first one can be solved by similarity between the questions have already answered and the question proposing now. Our model use TF-IDF algorithm to make words into vectors, specific formula is as follows [14].

$$\mathrm{TF}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$\mathrm{IDF}_i = \log\left(\frac{|D|}{\left|\{j : t_i \in d_j\}\right|}\right)$$

In the formula, i and j are the words in the questions, so we can use the product of TF and IDF to present the words in one question. Finally, we can get similarity of the new question and each question has already answered.

**(2) The new question:**

We display the TOP-five answers to users, if users choose one of these answers, the answering is end. If not, the answering transform to the second way, which is answering new questions. The normal intelligent QA system only contains the first part, however, there are new question entering the PKC platform continuously, so there should be some means to react this problem in the intelligent QA system of PKC platform. In the consequence, this part is the important contribution of our work.

In the process of answering new questions, we use the LSTM to predict the contents of the answers. The recurrent neural network language model (RNNLM) has shown significant promise for statistical language modeling [15]. Shi et al [16] also applied the LSTM whichadds doors in RNN to improve the accuracy of prediction. We use the LSTM with some features PKC platform has such as centrality and money. The content can be transformed into a words' vector $(w1, w2, w3 \ldots wn)$, so the problem is to predict the vector. The complete ways to get the vector is as follows:

$$(w1, w2, w3 \ldots wn) = LSTM(w1', w2', w3' \ldots wn')$$

The $wi'(i \in 1,2,3 \ldots n)$ is vector of $wi$, which represent the word in the answer. $wi'$consists of several elements. These elements are explained as follows:

Word features:

$f_{word-feature}(wi) = \big(entity(wi), TF(wi), previousword(wi)\big)$.

The $entity\ (wi)$ represents the entity of $wi$. If $wi$is entity, then the value is 1, if not, the value is 0. The $TF(wi)$represents the word frequency in this answer. The $previousword(wi)$represents the previous word of $wi$.

Word embedding:

$f_{word-embeddings}(wi) = \mathrm{E}(wi)$.

We can train the common words in advanced such as "hello", personal pronouns and so on because these words can influence the results of training on other words because of the word frequency.

Word experts' features:

$f_{word-expert}(wi) = \big(amountoflistening(wi), money(wi)\big)$.

The different aspects of PKC platform with the normal QA is the amount of listening and the money of the answers. The more amount of listening and much money mean that word is usually said by experts, so this word can be the probably answer for these questions.

Similarly, we can train the answers of each entity of corresponding questions as the methods above. However, the different of them is the word features in LSTM algorithm. There is $entity(wi)$ in word features of the process above but there is not in the learning process of the answers of each entity because the training set has the same entity in this learning process. The complete process of learning is displayed in Figure 3.

***3. Matching Module:***

We get the model trained through the Learning Module, in the module, we use the model to predict two kinds of results: one is the result of the learning process of answers of each category, the other is the result of the learning process of answers of each entity corresponding with the question. When we get these two kinds of results, we can get the all of words of them and remove the words irrelevant with questions, finally, the results are these suitable key words. The matching process of our model is displayed in Figure 3.

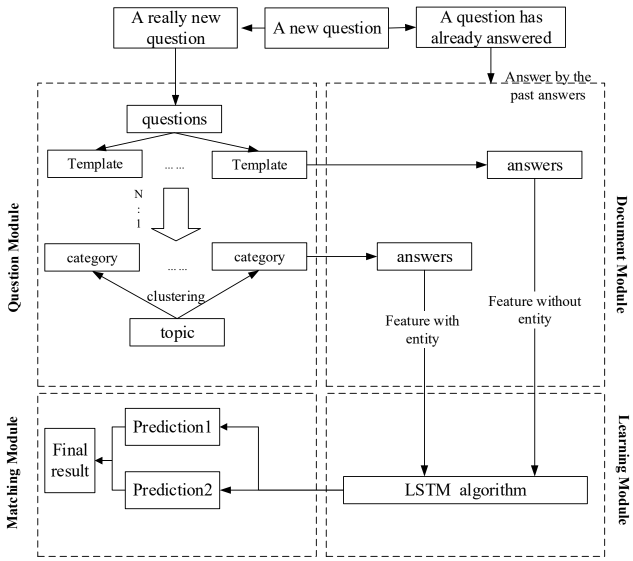


**Figure 3: The learning process in our model**

# 5 EXPERIMENTS

## 5.1 Data

In recent researches, the evaluation dataset is the mostly standard dataset such as SQuAD [17], CNN/Daily Mail [18]. Because the particularity of the PKC platform, we use the dataset of PKC platform.

We have collected the 1222 QA sites range from big data to movies and so on from a Chinese community. There are six topics in the dataset, every topic can be about all kinds of themes such as life, Internet and so on. The community is one of the best communities of persons who have professional knowledge, the questions where are normal, but the answers are logic and evidential in China. In the same time, in the community, there are many topics in this platform such as emotion, health, employment and so on. So, our data is comprehensive in topic and practical meaning in PKC market.

## 5.2 Results and discussion

We can design three indexes to evaluation our model, named Precision, Recall and F1-score [19]. The first one is exact matching of these key words predicted by our model, which means ratio of the number of key words completely appearing in the existing answer people have accepted widely and the number of words predictive answer has, we can define which as the Precision of our model. The second one is ratio of the number of key words completely appearing in the existing answer people have accepted widely and the number of words predictive answer has, we can define which as the Recall of our model. F1-score is the harmonic average of them. The detail formula is illustrated as follow.

$$\textbf{Precision} = \frac{the\ number\ of\ exact\ match\ words}{\text{the}\ number\ of\ words\ predictive\ answer\ has}$$

$$\textbf{Recall} = \frac{the\ number\ of\ exact\ match\ words}{\text{the}\ number\ of\ words\ best\ answer\ has}$$

$$\textbf{F1} - \textbf{score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

Considering the contents of PKC platform are developing with time, we choose the Question: "How to choose the cheapest commodities and are them really cheap in 'Double Eleven' (11st, Nov)?" as our experiment question. We use clustering method to define this question as the category of Internet. Because the question is new, so our model should answer this question by itself. Then the entity of this question is 'Double Eleven', and the training dataset is the QA sites of Internet category and the 'Double Eleven' entity. At the same time, to evaluate the accuracy of our model, we calculate the Precision, Recall and F1-scoreof the answer our model gets and make some comparisons.

To evaluate whether our model may fit the PKC platform more suitably, we choose the normal RNN algorithm as the compared model to our model. We can get the Precision, Recall and F1-score of these two models in different number of keywords, which includes 5, 10 and 20 keywords respectively. The detail results are displayed as Figure 4.

According to the results of comparison of normal RNN algorithm and our model, we find our model is better than normal RNN algorithm in the PKC platform. At the same time, we can find the Recall and the F1-score is decreasing as the number of key words increases, while the Precision remains stable as the number of key words increases. In the consequence, the high weight word is predicted more precisely in the experiment.

Then, we can compare the different results of different iterations of our model, the results are illustrated at Figure 5. We can find the results of 10000 iterations are best of them, although the Precision and F1-score of 40000 iterations are better in small number of key words, the integral prediction of which are less than another two ways. In the consequence, the more iterations may not lead to better results as the local optimum exists.

## 5.3 Improvement directions

In this paper, we just use experiments of two aspects to evaluate our model. And we just use one question as our evaluation question, which we can improve in the future studies.

First, we will try to answer more questions by our model, which means our model can train more data, these changes will improve the effectiveness of our model. Second, we may delete some information irrelevant with the question and some have much subjective meaning. Third, machine comprehension can be added into the implement of the Learning module, which may improve the results of learning and effectiveness of answering. Finally, the learning ratio of our Learning module can be changed to find the most suitable ratio, as according to the findings of this study, the relationship between indexes and iterations are still not obvious.

# 6. CONCLUSION

This paper aims to investigate how to apply the intelligent QA system into the PKC environment. Based on the results, we mainly have two findings: 1) The precision of our model in Question Answering can reach the level of normal answering in other CQA fields and our model is better than the normal RNN algorithm in the PKC platform. 2) The answering effectiveness can be improved by our model.

Our work also has some business implications: 1) Our work aims to improve the answering effectiveness of PKC platform, which is one of the most salient problems in the PKC platform. 2) We try to integrate intelligent QA systems and PKC platforms, and propose four modules of intelligent QA, which can be adopted by platform. 3) From cognitive computing aspects by combing machine learning and user behavioral theory, we investigates how to solve the QA problem, which may be adopted by researchers and practitioners to improve users' satisfaction from integrated and interdisciplinary perspectives.

The future works may focus on these aspects: 1) Take more experiments on the larger and real datasets. 2) Collect data from other PKC platforms to train the model so that we can compare the advantages and disadvantages of them. 3) Investigate how to input questions can have a more correct answer and what other features in the PKC platform we can use. 4) Explore whether this work can really enhance the business value of PKC, in other word and whether this work can improve the users' satisfaction or the benefits for company.

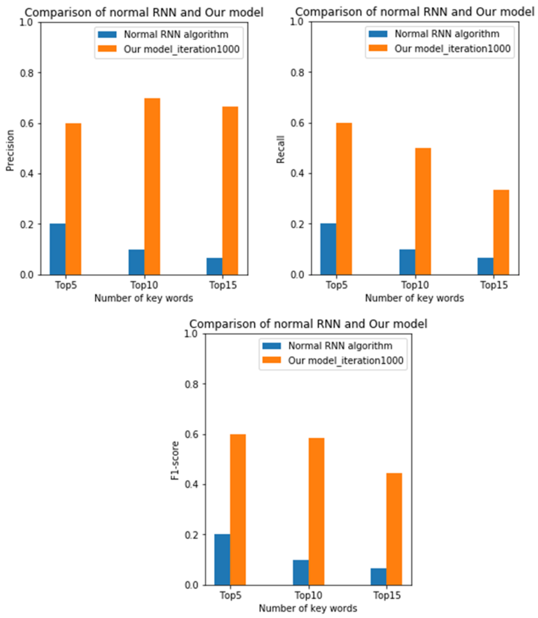


**Figure 4: Comparison of normal RNN and Our model**

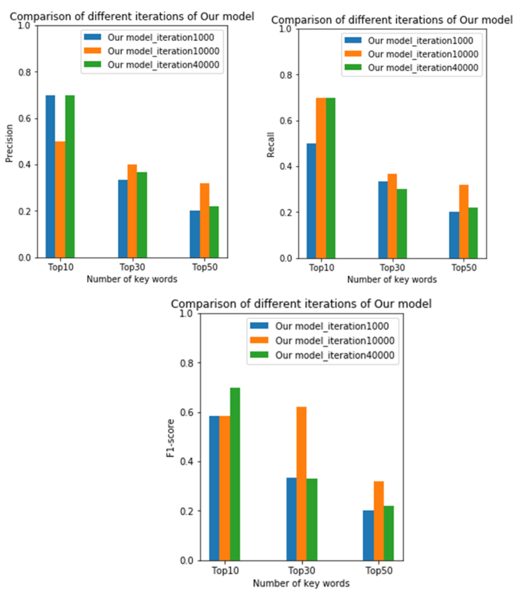


**Figure 5: Comparison of different iterations of Our model**

## REFERENCES

[1] Liu, Y., Bian, J. and Agichtein, E. Predicting information seeker satisfaction in community question answering. *International ACM SIGIR Conference on Research and Development in Information Retrieval* (2008), 483-490.

[2] Yang, L., Bao, S., Lin, Q., Wu, X., Han, D., Su, Z. and Yu, Y. Analyzing and predicting not-answered questions in community-based question answering services. *AAAI Conference on Artificial Intelligence* (2011), 1273-1278.

[3] Cui, W., Wang, H., Wang, H., Song, Y., Hwang, S. W. and Wang, W. KBQA: learning question answering over QA corpora and knowledge bases. *Proceedings of the Vldb Endowment*, 10, 5 (2017), 565-576.

[4] Agichtein, E., Liu, Y., & Bian, J. Modeling information-seeker satisfaction in community question answering. *Acm Transactions on Knowledge Discovery from Data*, 3, 2 (2009), 1-27.

[5] Li, B. and King, I. Routing questions to appropriate answerers in community question answering services. *ACM International Conference on Information and Knowledge Management* (2010), 1585-1588.

[6] Riahi, F., Zolaktaf, Z., Shafiei, M. and Milios, E. Finding expert users in community question answering. *Proceedings of the 21st International Conference on World Wide Web* (April 16 - 20, 2012 2012), 791-798.

[7] Zhang, J., Ackerman, M. S. and Adamic, L. Expertise networks in online communities:structure and algorithms. *International Conference on World Wide Web* (2007), 221-230.

[8] Yang, L., Qiu, M., Gottipati, S., Zhu, F., Jiang, J., Sun, H. and Chen, Z. CQArank: jointly model topics and expertise in community question answering. *ACM International Conference on Information & Knowledge Management* (2013), 99-108.

[9] Wang, Y. On Cognitive Computing. *International Journal of Software Science & Computational Intelligence*, 1, 3 (2012), 1-15.

[10] Chen, D., Fisch, A., Weston, J. and Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (July 30 - August 4, 2017 2017).

[11] Choi E, H. D., Uszkoreit J, et al Coarse-to-fine question answering for long documents. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* 1, 1 (2017), 209-220.

[12] He, S., Liu, C., Liu, K., Zhao, J., He, S., Liu, C., Liu, K. and Zhao, J. Generating Natural Answers by Incorporating Copying and Retrieving Mechanisms in Sequence-to-Sequence Learning. *Meeting of the Association for Computational Linguistics* (2017), 199-208.

[13] Hao, Y., Zhang, Y., Liu, K., He, S., Liu, Z., Wu, H., Zhao, J., Hao, Y., Zhang, Y. and Liu, K. An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge. *Meeting of the Association for Computational Linguistics* (2017), 221-231.

[14] Wu, H. C., Luk, R. W. P., Wong, K. F. and Kwok, K. L. Interpreting TF-IDF term weights as making relevance decisions. *Acm Transactions on Information Systems*, 26, 3 (2008), 55-59.

[15] Shi, Y., Zhang, W. Q., Liu, J. and Johnson, M. T. RNN language model with word clustering and class-based output layer. *Eurasip Journal on Audio Speech & Music Processing*, 2013, 1 (2013), 22.

[16] Shi, Z., Shi, M. and Li, C. The prediction of character based on recurrent neural network language model. *Ieee/acis International Conference on Computer and Information Science* (2017).

[17] Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (November 1-5, 2016 2016).

[18] Chen, D., Bolton, J. and Manning, C. D. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (August 7-12, 2016 2016).

[19] Hao, W., Yanmei, F., Qinyong, W., Hongzhi, Y., Changying, D. and Xiong, H. A Location-Sentiment-Aware Recommender System for Both Home-Town and Out-of-Town Users. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* ( August 13 - 17, 2017 2017), 1135-1143.