Data Driven Answer Retrieval in offline QA System

Rahul Narayan Nirmal[1], Samir Pratap Patil [2] , Avinash Ramesh Ghorpade[3] , Yogesh Kulbhushan Murkunde[4] , Prof. P.A.Tak[5]
[1, 2,3,4] **ZCOER**
[5] **Asst. Professor at ZCOER**

Abstract- *Searching similar questions from historical dataset has been applied to community question answering, with well theoretical underpinnings and great practical success. The searched question has more than one answers means we can get pool of multiple answers. Because of this it will take lot of time to browse all the pool of answers and go through it and choose the best one. To solve this problem in this paper we are providing the ranked answers in the form of pairwise comparisons. In particular, it consists of one offline and online learning component. In offline we can find the sentiment in positive, negative and neutral categories to find the proper rank of the answers and suggest best one in that. In this paper provide these three types of training samples. In the online search component, we first collect a pool of answer for the given question via finding similar questions. Then sort the answer candidates by leveraging the offline trained model to judge the orders preference. We have supported the real-time datasets and work on the offline and online methodology*

Keywords- Community-based Question Answering, Answer Selection, Observation-guided Training Set Construction.

I. INTRODUCTION

In this paper we are using questions and answers as input and novel Pairwise Learning method to RANK model called PLANE, which can quantitatively rank answer candidates from the relevant question pool. We use Offline learning and online search where in offline learning which is guided by our user studies and observations, where we automatically establish the positive, negative, and neutral training samples in terms of preference pairs from input. And when it comes to the online search, for a given question, we pair it with each of the candidate's answer, and fit them into the trained PLANE model to estimate their matching scores. And like wise we help to find best answer. When we try to find questions from QA systems we get lots of answers and to find what we want is very tough work because it consumes lots of time and we have to read each and every question manually so to solve this problem we are implementing this paper. Main objective of this system is to provide user with best and relevant answer for which user question he/she searched and this will save time of user and help them to get best information as quickly as possible. The success of cQA and participation of user, question starvation occurred in cQA forums, which refers to the following two kind of phenomena:
First, user usually has to wait long time getting answers to their question. For instance, a study over 300thousand questions in Quara. In short time complexity of getting right answer is very high.
Second, if any questions have no answer then it also takes time to get response. Considering Yahoo! Answers as an example, around 10% of its questions do not receive any answer and leave the askers unsatisfied
Due to the lack of an question routing mechanism, a user is easily overwhelmed by the large number of questions open, and cannot easily find questions user is interested in answering even if user contribute his/her knowledge. Thus, there is a serious gap between the existing open questions and potential answerers. To bridge the gap, we present a new approach to PLANE Model, which aims to get most effective answer for searched questions. From the seeker's perspective, it can reduce the time lag between the time a question is posted and the time it is answered, and it can potentially increase the asker's satisfaction to CQA services.

II. RELATED WORK

Jeon et al. [1] extracted a set of non-textual features it covers the contextual information of QA(Question Answers) pairs, and proposed a language model for processing features in order to predict the answers collected from a cQA service discusses A Ranking Approach on Large Scale Graph With Multidimensional Heterogeneous Information In this paper, we provides the large-scale graph-based ranking problem and focus on how to effectively discovers rich heterogeneous information of the graph to improve the ranking performance. Specifically, we propose an effective and innovative semi-supervised Page-Rank (SSP) approach to parameterize the derived information within a unified semi-supervised learning framework (SSLF-GR), then optimize simultaneously the parameters and the ranks scores of graph nodes.

Raikwal et al. [2] discuss the SVM and k-NN Algorithms for improving the quality of question answer system.

To work with SVM and K-NN we decide to perform complete task under three steps.

2.1 Experimental data retrieval : Different type of data selected as the experimental data set. To get the performance is varies or not according to data. Here we collect data of different size and different types, like we use data nominal data and numerical data both to evaluate results.

2.2 Data analysis model: Here the implementation of algorithms includes. Data analysis using different algorithm includes data analysis or model building using both data models.

2.3 Result: Different system generated resultant parameters are generated. Analysis of result includes the performance analysis of system on different parameters like memory uses, accuracy and search time.

A. Shtok et al.[3] Exploring heterogeneous features for query focused summarization of categorized community answers. is Based on the computed global ranking scores, we utilize two different strategies to construct top K candidate answer set, and finally solve a constrained optimization problem on the sentences to top K answers to generate as summary towards a user's query.

III, ANSWER RANKING:

In case of directly rank community answers, some researcher's resorts to identify user's authority via graph-based link analysis. The techniques a graph-based link analysis have been well-studied in the social network analysis and hence achieved goal [4], [5], [6]. In the Question-Answer task, they assumed that the authoritative users tend to generate high- quality answers [7].

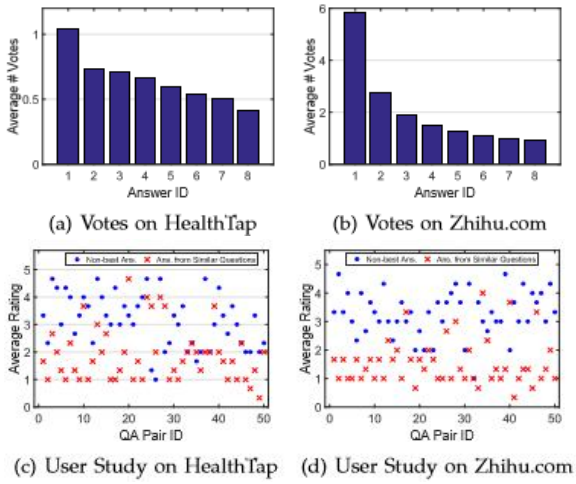

Fig.1. Subfigures (a) and (b) illustrate the number of vote distributions over HealthTap and Zhihu.com, respectively. The answers of each question were sorted decreasingly regarding their votes in advance. Subfigures (c) and (d) respectively display the user study results of QA match over HealthTap and Zhihu.com.

IV. OFFLINE LEARNING:

To gain the insights into the answer quality in QA, we collected a dataset of questions and their answers from various Question Answers, given website HealthTap, yahoo answers.com, stackoverflow.com, and the general one zhihu.com, respectively. For each question, we sorted its answers in decreasing order regarding the number of "votes". Hereafter, we counted the average number of votes over all the answers ranked at the same positions. From fig 1(a),1(b) we illustrate following observation.

For a search question, its best answer is preferable to its non-best answers. In particular, for each search question, we found that its best suitable answer is always positioned at the first place in terms of votes. Furthermore, on average, the votes of the best answers far outnumber those of the rest.
The non-best answers of a question are almost on a par. Regarding the non-best answers, we cannot see a significant "vote" drop between two successive ranks

A question prefers the answers of itself to those of others. We observed this point from a user study. In particular, we randomly selected 50 questions from our collected HealthTap and Zhihu.com datasets, respectively. For each question, we provided two answers: one was randomly selected from its non- best answers, and the other was randomly selected from those of its similar questions calculated via k- nearest neighbors (k-NN).

V. SYSTEM ARCHITECTURE

5.1 Exiting System:-

Earlier in QA system where questions were asked obtaining answers took lot of time and answer which answer we get is not what we want and the process is time consuming and it is not like when we send questions. Different methods where used to rank sort and divide answers. The main objective is to provide users to get information from CQA systems where lots of discussion is done and to find out information we are providing this system. CQA systems have lots of information and this can be used for understanding and finding out information on topics. We use already available data from CQA systems and train our system to get output. However, there can be information on topics which are not

available in that dataset so we can add them later for other topics.

Following diagram shows the existing system of cQA System

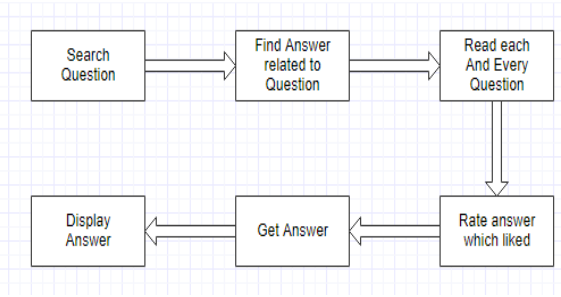

Fig 1. Exiting System

5.2 Proposed PLANE Model:

Given a question, we can quickly obtain a set of top k relevant questions Q = {q1,...,qk} from the archived QA repositories via the well-studied question matching algorithm k-NN. Without loss of generality, we assume question qi has a set of m_i≥1answers, denoted $A_i = \{a_i^0, a_i^1, \ldots, a_i^{m_i}\}$, whereby 0 i is the best answer of qi selected by community users. We aim to develop a learning to rank model to sort all the answers associated to the return edrelevant questions in Q.

As discussed previously, given a set of QA pairs, we can build the dual training sets X and U. To jointly incorporate X and U, we propose the following pairwise learning to rank model,

$$\min_{\mathbf{w}} \sum_{i=1}^{N} [1 - y_i \mathbf{w}^T \mathbf{x_i}]_+ + \lambda \|\mathbf{w}\|_1 + \mu \sum_{j=1}^{M} |\mathbf{w}^T \mathbf{u}_j|,$$

The rest term is a hinge loss function, which is suitable for our binary preference judgment task. It provides a relatively tight and convex upper bound on the 0-1 indicator function. Besides, the empirical risk minimization of this loss is equivalent to the classical formulation for support vector machine (SVM) . Correctly classified points lying outside the margin boundaries of the support vectors will not be penalized, whereas points within the margin boundaries or on the wrong side of the hyper plane will be penalized in a linear fashion compared to their distance from the correct boundary. The second term is a ℓ1 norm, which regularizes w and helps in feature selection. The last term is a sum of absolute values, which aim to penalize the preference distances between non-best answers of the same questions, and it guarantees our second observation in nature

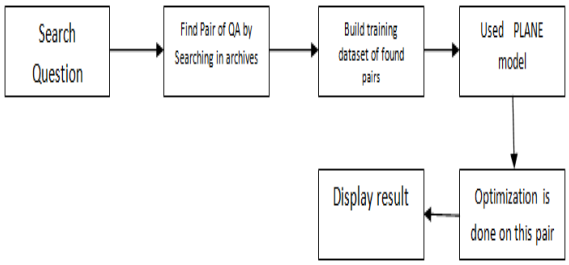

Fig 2. Propose PLANE Model

Search a question, instead of choosing the best answer from the most relevant question, in this paper, we present a novel Pairwise Learning to rank model, nicknamed PLANE, which can quantitatively rank answer candidates from the relevant question pools. In that two components are present: online search and offline learning. Particularly, during the offline learning calculate the sentiment positive, negative, and neutral training samples in terms of preference pairs. The PLANE model can be jointly trained with these three kind of training samples. We conduct extensive experiments over two datasets, collected from a vertical CQA site Zhihu.com and a general CQA site HealthTap, respectively

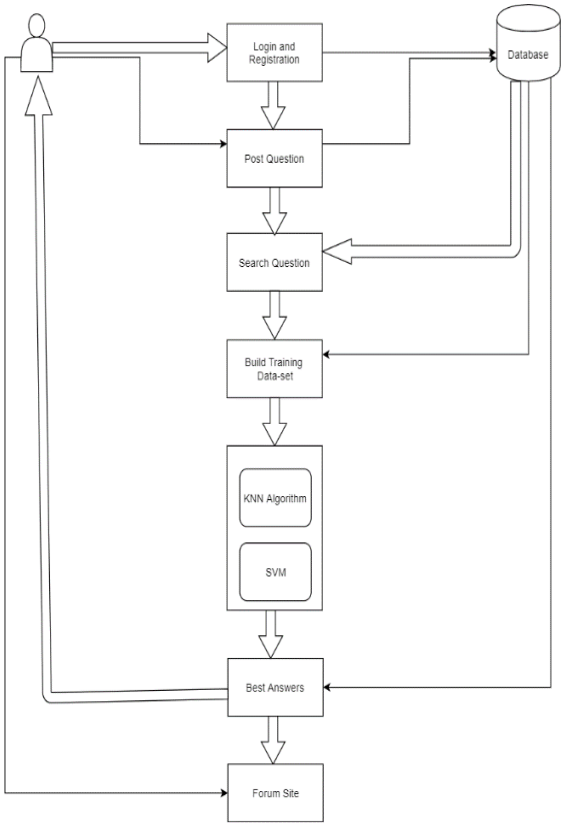

Fig 3 Architecture of Proposed System

5.2.1 SVM

The support vector machine has been chosen because it represents a framework both interesting from a machine learning perspective. A SVM is a linear or non-linear classifier, which is a mathematical function that can distinguish two different kinds of objects. These objects fall into classes, this is not to be mistaken for an implementation To work with SVM we use leaner kernel for implementation. In functional Analysis and linear algebra, the kernel of a linear operator L is the set of all operands v for which L(v) = 0. That is, if L: V → W, then

ker(L) = { v € V : L(v)=0 }

The Ranking SVM (RankSVM) algorithm is a learning retrieval function that employs pairwise ranking methods to adaptively sort results based on how 'relevant' they are to a specific query [8]. The original purpose of the algorithm was to improve the performance of an internet search engine. Hieber et al. [9] used RankSVM to improve the performance of answer ranking in social QA portals and achieved promising performance. Similar as GBRank, the model was trained with the answer pairs under the same question.

Algorithm:

initialize yi = YI
where yi is keywords in question for i ∈ I REPEAT compute SVM solution w, b for data set with keywords where b is bag compute outputs fi = where fi is distance between keywords, xii + b are for all xi in positive bags set yi = sgn(fi) where xi is keywords in bag for every i ∈ I, YI = 1
FOR (every positive bag BI)
IF (words == yi) compute i ∗ = arg max i∈I words
set yi* = 1 END
END
WHILE (keywords have changed)
OUTPUT (w, b)

Svm is used to find best answers from relevant found answers it finds best answer and it helps to train system to find best answers.

5.2.2. k-NN:

In pattern recognition or classification, the k-nearest neighbor algorithm is a technique for classifying objects based on closest training examples in the problem space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification [3]. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. The k-NN algorithm can also be adapted for use in estimating continuous variables. One such implementation uses an inverse distance weighted average of the k-nearest multivariate neighbors. The K-nearest-neighbor (KNN) algorithm measures the distance between a query scenario and a set of scenarios in the data set.
Distances
We can compute the distance between two scenarios using some distance function , where are scenarios composed of features, such that. Two distance functions are discussed in this summary:

Absolute distance measuring:

$$d_A(x,y) = \sum_{i=1}^{N} |x_i - y_i|$$

Euclidean distance measuring:

$$d_E(x,y) = \sum_{i=1}^{N} \sqrt{x_i^2 - y_i^2}$$

Because the distance between two scenarios is dependent of the intervals, it is recommended that resulting distances be scaled such that the arithmetic mean across the dataset is 0 and the standard deviation 1. This can be accomplished by replacing the scalars with according to the following function:

$$x' = \frac{x - \bar{x}}{\sigma(x)}$$

Where is the unsealed value, is the arithmetic mean of feature across the data set is its standard deviation and is the resulting scaled value.

Algorithm:

Classify(X,Y,x);(X is training dataset, Y is keywords, x is input keywords)
for I=1 to n
Compute distance d(Xi,x)
End for
Compute set of keywords where k is smallest distances for(Xi,x)
Return Indices of least distances.

VI. RESULT

We expected best and relevant answers for searched questions by user on CQA system. we implemented this system on windows and used jsp servlets and used jdbc

database .And this system we used archives of answers and used those to find questions.

VII. CONCLUSION

In this paper, we are providing new way to find best and relevant answers for asked questions. It supports with two online and offline components where in offline we train our system based on asked question and find answers based on it. In offline we calculate the create training samples in the forms of preference pairs using keywords in question. In the online search component, for a given question, we first collect a pool of answer candidates by finding its similar questions using plane model where we rank answers based on question and when user search he will be given best and relevant answer and then he can rate answers so that next time user will get that rated answer at top.

REFERENCES

[1] J. Jeon, W. B. Croft, J. H. Lee, and S. Park, "A framework to predict the quality of answers with non-textual features," in Proceedings of SIGIR'06. ACM, 2006, pp. 228–235. 2

[2] J.S.Raikwal, performance evaluation of svm and knn algorithm over medical dataset, ser IJCA(0975-8887), 50 – No.14, July 2012

[3] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor, "Learning fromthe past: Answering new questions with past answers," in Proceedings WWW'12. ACM, 2012, pp. 759–768. 1, 3

[4] X. Li, M. K. Ng, and Y. Ye, "Multicomm: Finding community structure in multi-dimensional networks," TKDE, vol. 26, no. 4, pp. 929–941, 2014. 3

[5] W. Wei, G. Cong, C. Miao, F. Zhu, and G. Li, "Learning to find topic experts in twitter via different relations," TKDE, vol. 28, no. 7, pp. 1764–1778, 2016. 3

[6] W. Wei, B. Gao, T. Liu, T. Wang, G. Li, and H. Li, "A ranking approach on large-scale graph with multidimensional heterogeneous information," TOC, vol. 46, no. 4, pp. 930–944, 2016. 3

[7] X.-J.Wang, X.Tu, D.Feng, and L.Zhang,"Ranking community answers by modeling question-answer relationships via analogical reasoning," in Proceedings of SIGIR'09. ACM, 2009, pp. 179–186. 3

[8] F. Hieber and S. Riezler, "Improved answer ranking in social question-answering portals," in Proceedings of SMUC'11. ACM, 2011, pp. 19–26.

[9] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon, "Adapting ranking svm to document retrieval," in Proceedings of SIGIR'06. ACM, 2006, pp. 186–193.

[10] Z.Ji and B. Wang, "Learning to rank for question routing incommunity question answering," in Proceedings of CIKM'13.ACM, 2013, pp. 2363–2368. 2

[11] T. C. Zhou, M. R. Lyu, and I. King, "A classificationbasedapproach to question routing in community questionanswering," in Proceedings of WWW'12. ACM, 2012, pp. 783–790. 2

[12] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen, "Cqarank: Jointly model topics and expertise incommunity question answering," in Proceedings of CIKM'13.ACM, 2013, pp. 99–108. 2

[13] B. Li and I. King, "Routing questions to appropriate answerersin community question answering services," in Proceedings ofCIKM'10. ACM, 2010, pp. 1585–1588. 2

[14] K. Wang, Z. Ming, and T.-S. Chua, "A syntactic tree matching approach to finding similar questions in community-based qaservices," in Proceedings of SIGIR'09. ACM, 2009, pp. 187–194.2, 6

[15] Y. Liu, J. Bian, and E. Agichtein, "Predicting information seeker satisfaction in community question answering," in Proceedings of SIGIR'08, ser. SIGIR '08.ACM, 2008, pp. 483–490. 2

[16] M. J. Blooma, A. Y. K. Chua, and D. H.-L.Goh, "A predictive framework for retrieving the best answer," in Proceedings ofSAC'08. ACM, 2008, pp. 1107–1111.

[17] L. Nie, M. Wang, Y. Gao, Z. Zha, and T. Chua, "Beyondtext QA: multimedia answer generation by harvesting webinformation," TMM, vol. 15, no. 2, pp. 426–441, 2013. 3

[18] Q. H. Tran, V. Duc, Tran, T. T. Vu, M. L. Nguyen, andS. B. Pham, "Jaist: Combining multiple features for answer Selection in community question answering," in Proceedings ofSemEval'15. ACL, 2015, pp. 215C–219. 3

[19] W. Wei, Z. Ming, L. Nie, G. Li, J. Li, F. Zhu, T. Shang, andC. Luo, "Exploring heterogeneous features for query-focused Summarization of categorized community answers," Inf. Sci.,vol. 330, pp. 403–423, 2016. 3

[20] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton,"Quantitative evaluation of passage retrieval algorithms forquestion answering," in Proceedings of SIGIR'03. ACM, 2003,pp. 41–47. 3

[21] H. Cui, R. Sun, K. Li, M.-Y.Kan, and T.-S. Chua, "Questionanswering passage retrieval using dependency relations," in Proceedings of SIGIR'05. ACM, 2005, pp. 400–407. 3

[22] R. Sun, H. Cui, K. Li, M.-Y. Kan, and T.-S. Chua, "Dependencyrelation matching for answer selection," in Proceedings of SIGIR'05. ACM, 2005, pp. 651–652. 3

[23] M. Surdeanu, M. Ciaramita, and H. Zaragoza, "Learning torank answers on large online qa collections," in Proceedings ofACL'08. ACL, 2008, pp. 719–727. 3

[24] A. Agarwal, H. Raghavan, K. Subbian, P. Melville, R. D.Lawrence, D. C. Gondek, and J. Fan, "Learning to rank forrobust question answering," in Proceedings of CIKM '12. ACM,2012, pp. 833–842.

[25] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne,"Finding high-quality content in social media," in Proceedings of WSDM'08. ACM, 2008, pp. 183–194. 2

[26] L. Nie, M. Wang, L. Zhang, S. Yan, B. Zhang, and T. S. Chua,"Disease inference from health-related questions via sparse deep learning," TKDE, vol. 27, no. 8, pp. 2107–2119, 2015. 1