3M1-IOS-3a-2

Question Routing by Modeling User Expertise and Activity in cQA services

Liang-Cheng Lai and Hung-Yu Kao

Department of Computer Science and Information Engineering

National Cheng Kung University, Tainan, Taiwan, R.O.C

Community Question Answering (cQA) sites such as Yahoo! Answers and Stack Overflow have emerged as a new type of community portal that allows users to answer the questions asked by other people. The cQA archives have accumulated a huge mass of questions and answers. On account of the progressively increasing questions, there are numbers of questions to be solved or answered by others. In this paper, we address the problem of efficient question routing. We present a new approach that combines user's expertise and user's activity to solve this problem. First, we model user's expertise by the contents of user's answering questions in the past, and then we enhance user's expertise by social network characteristic in the cQA portal. Second, we model and predict user's activity by analyzing the distribution of their previous answering records. Experiments conducted on a real cQA data, Stack Overflow, show that our approach leads to a better performance than other baseline approaches significantly. In terms of the evaluation metric, MRR, the performance of the content-based approach is 0.0999 and that of our method is 0.1372 respectively. We can get a 37.34% improvement over the traditional content-based method. On average, each of 6,160 test questions gets at least one answer if it is routed to the top 7 ranked users by our approach.

1. Introduction

Recently, Community Question-Answering (cQA) has already attracted a great deal of attention from researchers. In cQA sites, people can share or acquire knowledge and information from these systems. Yahoo! Answers (http://answers.yahoo.com), Live QnA (http://qna.live.com), and Stack Exchange (http://stackexchange.com/) are examples of these forums. These portals have attracted increasing number of users and accumulated a huge mass of questions and answers over the past few years. Through these portals, users are allowed to answer the questions asked by other people, or users are allowed to pose their information need as questions.

On account of the progressively increasing questions, there are numbers of questions which cannot be solved or answered by others efficiently. To solve this problem, Zhou et al. proposed the push mechanism in cQA services [Zhou 2009]. This mechanism helps push new questions to proper users to get answers. New questions are routed to appropriate users with enough knowledge and great activity to be solved effectively and efficiently. In our work, we focus on this problem and propose our approaches to support and augment this mechanism.

In order to investigate whether new questions can be solved efficiently in cQA services, we observed the data from the cQA site, Stack Overflow, in the duration of October 2010. **Figure 1** shows the time interval of receiving answers per solved question. Our goal is to solve the questions in the dotted block of **Figure 1** and other unsolved questions during this period. There are totally 71,000 new questions are posted in this month. We found that only 60,403 questions have been answered in this month. That is, there are 10,597 (15%) questions which are not answered by other users in this month. Furthermore, only 48,055 questions are answered in 2 hours. There are still 22,945 (32.3%) questions which cannot be solved efficiently. On average, users who post questions in this cQA website have to wait at least 8.12 hours to receive answers. Because of so many unsolved questions and long waiting time, an effective push mechanism is needed in cQA portals. To develop the push mechanism, we address the problem of effectively finding the potential users with enough knowledge and great activity to answer the new questions in cQA services.

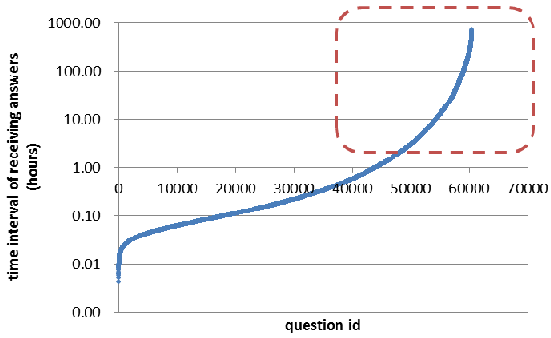

Figure 1: Time interval of receiving answers per solved question.

Over the past few years, cQA services have attracted many users and accumulated numbers of questions and answers. Considering the scale of online users in cQA, it is a nontrivial job to route a new question to appropriate answerers who are able to answer it quickly. Two simple approaches for the push mechanism are described as follows: *the content-based method* and *the QA-relation based method*.

The content-based methods route a new question to a user according to his answering questions in the past. The content similarity between the questions users answered and the new question is considered. If the questions user answered in past days are similar to the new question, we route this new question to this user. However, Bag-of-Word approaches cannot detect such these similar questions. Furthermore, the length of questions is relatively short comparing to web pages. Using the traditional

Contact: Hung-Yu Kao, CSIE, National Cheng Kung University, No.1, Ta-Hsueh Road, Tainan, Taiwan, R.O.C., +886-6-275757, +886-6-2747076, hykao@mail.ncku.edu.tw

content based search algorithm such as Okapi BM25 algorithm or language model [Zhai 2004] may not work well because of the short length of contents of questions. For example, we observed the answering questions of the user u_{405015} in Stack Overflow. This user has answered six questions as shown in **Table 1** (See Appendix for contents of the answering questions). We can see from this table that the titles and the contents of these questions have few same words even if they are topically related. If we adopt traditional content-based methods to determine whether new questions are suited to users or not, it may not work in this example.

Table 1: Titles of answering questions of User u_{405015} in Stack Overflow dataset.

Question id	*Title of question*
$q_{6618485}$	*CSS: Width 100% is not 100% of Screen*
$q_{6614999}$	*Fixed inline-block div with negative margin-right and shifting float: what's special about -4px?*
$q_{6614886}$	*A span can be a div, but a div can't be a span*
$q_{6600523}$	*Internet Explorer Specific CSS Glitch*
$q_{6598083}$	*Table not resizing properly in IE7*

The QA-relation based methods utilize a special structure in cQA services, i.e., the question-reply relationship. **Figure 2**(a) shows the relation between users and questions while **Figure 2**(b) shows the relation between users and users. Considering the problem of question routing, new questions are needed to route to people with high expertise. Relying on the content has proven to be limited to rank users' expertise levels [Littlepage 1997], previous works [Campbell 2003, Dom 2003] have shown that using graph-based algorithms can be more effective than using content-based methods alone. There have some work adopt the graph-based ranking algorithms such as PageRank [Brin 1998] or HITS [Kleinberg 1999] algorithm to rank users [Jurczyk 2007, Zhang 2007] by their expertise. However, contents of questions are not considered in these algorithms. It is not appropriate to route questions of different topics to the same users with high authority values.

Considering the example of question-reply graph illustrated in Figure 3, the labels of edges represent the topics of questions. If now we have one new question about "perl", this question would be routed to User 4 according to the result by graph-based algorithms because of its higher in-degree. However, we should route this question to User 3 because the answering questions of User 3 are more similar to the topics of "perl" in the past.

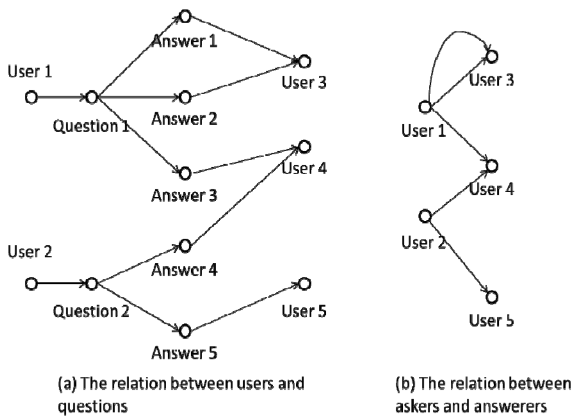

Figure 2: The QA relationship in cQA.

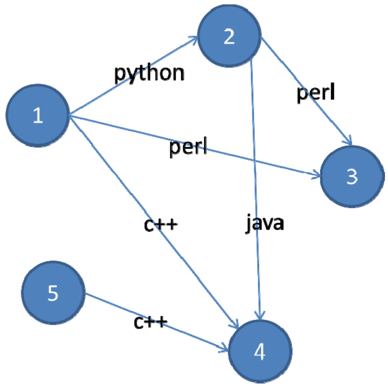

Figure 3: An example of question-reply graph.

Because of the shortcomings of directly applying the concepts of content and QA-relations, we cannot just apply these two basic methods to solve our problem of question routing.

Previous works [Li 2010, Zhou 2009] of Question Routing (QR) in cQA services focus on the contents of questions that users answered in their answering history. Zhou et al.'s [Zhou 2009] apply the three models to compute the user's expertise, namely a profile-based, a thread-based, and a cluster-based model. The profile-based model considers the user's answering questions as a group and the thread-based model treat user's answering questions individually while the cluster-based model clusters the questions into groups. In Li's work [Li 2010], questions are routed to the users with high expertise and availability. This work combines content-based approach with the work of answer quality issue in cQA services. Answering more questions similar to the new question with good quality, users are more likely to answer this new question. These two works do not consider the activity of users. If we route a question to a user who is inactive even though this user is able to answer this question, we cannot get new questions solved or get the answer efficiently. Hence, we should route new questions to the users not only with great expertise, but with high activity. The information for a specific user we have is less and the data we can use is only the past QA information of this user. Using these data to predict user's activity is difficult because of irregularity and insufficiency of information.

To address the question routing problem, we propose a framework to find the proper users for a given question to get new questions solved effectively and efficiently in cQA services. Based on the properties we mentioned in previous sections, we address the shortcomings of two baseline approaches by combining content-based approaches and QA-relation based approaches. Furthermore, we utilize the time characteristic of each user since user's activity is also indispensable in the question routing problem.

First, we compute the expertise score of users according to the contents of user's answering questions and social network characteristic as illustrated in **Figure 4**. We then estimate the expertise score of each candidate user for a given question based on the contents of questions in answering history of this user. We also utilize the question-reply graph to enhance the expertise score of each user by peer-expertise dependency. We define the peer-expertise dependency as a mutual reinforcement relationship between the asking and the answering of users. It assumes that expert users of a question q are those users who have answered many other questions (denoted as Q) that are related to q, and these questions Q are asked by those users who have great authority values. Finally, we combine the content based approach and peer-expertise based approach to measure the overall expertise quantity of each user.

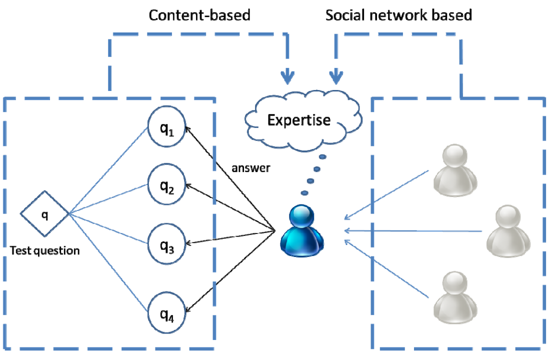

Figure 4: Illustration of the proposed expertise model.

Second, we compute the activity score of each candidate user. We estimate the activity score of each user based on their answering records from a period of answering history. In addition, we observe that most questions get only one answer and the response time of questions is usually short. It's important for users to anticipate others answering new questions. Thus we consider the active time of users per day and propose modeling the daily activity of users.

Finally, we estimate user's expertise and activity and combine these two models to rank users for each test question.

2. Related Work

The task of question routing is related to question retrieval, answer quality issue, and expertise ranking in social communities. In recent years, question retrieval has attracted much attention in research areas recently. Users directly search from the QA archive to find similar questions with respect to their questions when they want to seek knowledge. Therefore, the retrieval task in cQA services is the task of finding relevant similar questions with new queries. Consider the problem we want to solve in this paper, when a new question is posted, we want to know whether this question suits the user or not. That is, if a user answered a lot of questions similar to the test question, he might be able to answer this new question. We address the problem of question routing by the idea of question retrieval. We view the questions in user's history as the question database and new questions as new queries respectively. The goal is to extract the similar questions matching new questions in user's history to weigh the strength of users and new questions.

Jeon et al. proposed the translation-based model to find semantically existing similar questions from a community QA portal [Jeon 2005]. Their method can detect semantically similar questions even if they have little word overlap. The reason is that they calculate the question-question similarities by using the corresponding answers as well as the questions. In another work, Cao et al. [Cao 2008] address the problem of question recommendation by the following steps: First, questions are represented as graphs of topic terms, and the recommendations are then ranked on the basis of the graphs. Second, MDL-based (Minimum Description Length) is employed for selecting the best cuts. Question features are replaced with other features around the same question topics. Wang et al. considered the structure of question sentence and used the syntactic tree to match two questions [Wang 2009]. The basic intuition behind this is that if two questions' structures resemble each other, and they would be similar to each other with higher probability. They propose a new retrieval framework based on syntactic tree structure to tackle the similar question matching problem. **Figure 5** shows an example of a syntactic tree of the question "How to lose weight?". Their experimental results revealed that using the information of question's structure can significantly improve the similar question matching performance. Qu et al. modified Probabilistic Latent Semantic Analysis (PLSA) to model the relationships between users and questions for question recommendation and propose a novel metric to evaluate the performance [Qu 2009]. Questions are recommended to a user according to the probabilities of those questions given the user. Wu et al. also modified PLSA algorithm to propose the incremental recommendation algorithm, which considers not only the users' long-term and short-term interests, but also users' negative and positive feedback [Wu 2008].

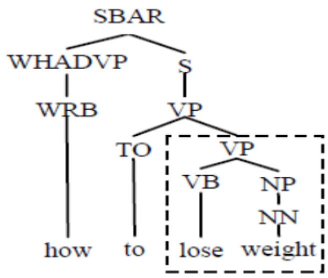

Figure 5: An example of syntactic tree of the Question "How to lose weight?".

Our work is also related to finding good quality answer with respect to the question in cQA services. As mentioned in section 1, if one user answered many questions similar to the new question, he might have higher probability to answer this new question. However, the quality of their answers is not considered. Users providing answers with good quality could be a strong evidence to be the expert for a given new question.

Jeon et al. uses a classification approach to identify the answer to be good or bad to address the answer quality problem [Jeon 2006]. He extracted several non-textual answer features in Naver (http://naver.com/), a Korean question/answering portal similar to Yahoo! Answers. These features include answer length, answerer's number of answers, and answer rating etc., for training and predicting the quality for the given answer. After extracting the features of each answer, non-monotonic features are converted to monotonic features and Maximum Entropy is used for answer quality estimation. Agichtein et al. derive multiple answer features from the graphs including structural, textual, and community features and they then use these feature to build the classification model to predict whether the answer is good or not [Agichtein 2008]. The contributions of the different sources of quality evidence are studied, and some of the sources are complementary. Combination of multiple types of sources is helpful to increase the performance of the system and increase the classifier's robustness to spam. Suryanto et al. used a graph-based answer ranking models by HITS algorithm [Suryanto 2009]. They combined a link-based and content-based approach to find the answers with good quality and relevant to the given question. There are four expertise based methods presented in this work. According to the result of this work, methods using question dependent expertise including the expertise of asker and answerer have the best performance.

Expert finding has attracted much research attention in recent years [Jurczyk 2007, Zhang 2007]. Zhang et al. [Zhang 2007] ranks users by applying link-based algorithm including Hyperlink-Induced Topic Search (HITS) [Kleinberg 1999] or ExpertiseRank in the website "Java Forum" based on users' authority scores.

HITS algorithm is used to rank web pages originally. Web pages are ranked by analyzing their inlinks and outlinks. In this algorithm, authorities mean the web pages pointed to by many hyperlinks while web pages which point to many hyperlinks are called hubs. Authorities and hubs are mutual reinforcing as: an authority pointed to by several nodes with high hubs should be a strong authority. A hub which points to many nodes with high authority should be a popular hub. The scores of hubs and authorities are defined as follows:

$$autho(p) = \sum_{q \in I(p)} hub(q) \quad (1)$$

$$hub(p) = \sum_{q \in O(p)} autho(q) \quad (2)$$

where $autho(p)$ and $hub(p)$ indicate the hub score and the authority score of page p. $I(p)$ represents the set of pages pointing to page p; $O(p)$ represents the set of pages pointed be page p.

Such link-based algorithms are applied in a bipartite network where an asker is linked to an answerer when a question posted by the former has been answered by the later. In another work [Jurczyk 2007], a similar approach is applied on the dataset crawled from Yahoo! Answers. From these proposals' results, we can get a conclusion that there is a high correlation between link-based metrics and user's expertise. However, in our problem of question routing, if we return a user ranking based on the authority score, we always get the same user ranking. As will be shown in experimental results, we use PageRank as one of the baselines to compare. In our work, we consider the content of test question to get the user ranking more precisely.

In other social media, Balog et al. propose generative probabilistic models to address the expert finding problem in enterprise corpora [Balog 2006]. Two general strategies to expert searching given a document collection are presented. The first one models an expert's knowledge based on the documents that they are associated with while the second one finds experts by locating the topics of documents. From 2005, Text REtrieval Conference (TREC) has provided a platform with the Enterprise Search Track for researchers to empirically assess their methods for expert finding [Craswell 2005]. In addition, there are some researches for finding experts over the e-mail corpus such as [Campbell 2003, Dom 2003]. Graph-based ranking algorithms are applied to rank email correspondents according to their expertise on subjects of interest. In the graph, nodes represent the correspondents and edges mean the relation of email correspondence between nodes respectively.

Language models have strong foundations in statistical theory and have performed quite well in many information retrieval tasks [Ponte 1998, Zhai 2004]. Typically language model approach can be divided into two types: profile-based and document-based methods. The profile-based approach (e.g., [Balog 2006]) estimates the probability of a candidate being an expert given the query topic by modeling the knowledge of an expert from associated documents. While in document-based methods, [Balog 2006] finds relevant documents for a given topic first. Then it ranks the candidates based on these documents. In our work, we use the language models to estimate the strength between users and questions.

We describe other related work in this sub-section. The push mechanism can not only help improve the performance of cQA services, but fulfill user's satisfaction and information need. Liu et al. [Liu 2008] also want to improve user's satisfaction (especially askers) by solving the problem of predicting information seekers satisfaction in cQA services. A general prediction model is presented and they develop a variety of content, structure, and QA based features.

There are lots of research studies about the analysis of cQA system. Adamic et al. [Adamic 2008] analyze the forum categories and cluster them based on content characteristics and patterns of user interaction to understand the knowledge sharing activity in Yahoo! Answers. The findings showed that different categories may have different characteristics. For example, interactions in users in some categories resemble expertise sharing forums while in other categories may represent the incorporate discussion, advice or support etc. Gyongyi et al. [Gyongyi 2007] analyzed 10 months worthy of Yahoo! Answers data to discuss the user behavior and impact in cQA portals.

3. Method

In this section, we present our proposed framework to address the problem of question routing to appropriate users in cQA services. Figure 6 shows the framework of our approach. Our framework consists of two components: the expertise model and the activity model. The expertise model ranks users according to their expertise calculated by content-based model and peer-expertise model. The content-based model calculates the similarity between user's answering questions and test questions as user's expertise of test questions, and the peer-expertise model calculates user's authority as user's expertise through the question-reply relation in cQA services. The activity model analyzes user's answering distribution and estimates the activity score of each user at the time of the routed question. The Linear Regression model combines the expertise index and activity index to be the user's final score of each test question. We define the combined score as *Expertivity* score, donated as $Expertivity(q,u)$.

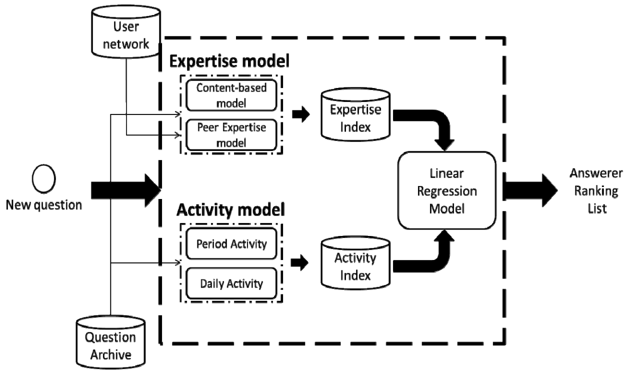

Figure 6: The proposed framework.

3.1 Expertise model

Given a test question q, we determine the expertise score of a user u by content-based model and peer-expertise model. In the content based model, we utilize the same approach as illustrated in [Li 2010]. We use this method to model user's expertise according to the content similarity between the answering questions of users and test questions. If the test question is similar to the questions answered by the user in the past, he/her would have more probability and ability to answer this question. Hence we use content-based model to model the strength between users and questions. The other component of expertise model is peer-expertise model. We employ and adjust the traditional PageRank algorithm and run the modified algorithm on the weighted question-reply graph. Then we get the authority value of each user for each test question.

QLL: Query Likelihood Language

We use the query likelihood language (QLL) model as a similarity measure to weigh the strength of users and questions. For a new question q to be routed, we define the score of user u as equation (3).

$$\boldsymbol{QLL(q,u) = P(q|q_u)} \quad (3)$$

$$\boldsymbol{P(q|q_u) = \prod_{w \in q} P(\omega|q_u)} \quad (4)$$

Let q_u denote all previously answered questions by user u. The meaning of this equation means how likely the new question q can be generated from the questions answered by user u. Since many words of question q may not appear in the contents of answering questions of users, we apply Jelined-Mercer smoothing method [Zhai 2004] to avoid assigning zero probability $P(\omega|q_u)$. The probability smoothed is defined as equation (5).

$$\boldsymbol{P(\omega|q_u) = (1-\lambda)p'(\omega|q_u) + \lambda p(\omega|Q)} \qquad (5)$$

where $p(\omega|Q)$ denotes the background language model built by the entire questions Q and λ is a coefficient controlling the influence between the background model and the probability estimated from the answering question of the user u. $p'(\omega|q_u)$ and $p(\omega|C)$ are defined as (6) and (7) respectively through a maximum likelihood estimation.

$$\boldsymbol{p'(\omega|q_u) = \frac{tf(\omega, q_u)}{\sum_{\omega' \in q_u} tf(\omega', q_u)}} \qquad (6)$$

$$\boldsymbol{p(\omega|Q) = \frac{tf(\omega, Q)}{\sum_{\omega' \in C} tf(\omega', Q)}} \qquad (7)$$

where Q is the collection of all questions in the database and $tf(\omega, q_u)$ represents how many times user u uses the term ω. $tf(\omega, Q)$ means frequency of the term ω appearing in the entire collection.

PeER: Peer-Expertise Rating

In this sub-section, we use peer-expertise model to calculate user's expertise score, $WPR(q,u)$. We utilized and augmented the idea of Weighted PageRank [Xing 2004] by question-reply relationship. The assumption of this algorithm is described as follows: Expert users of question q are those users who have answered many other questions that are related to q asked by those users with great authority values.

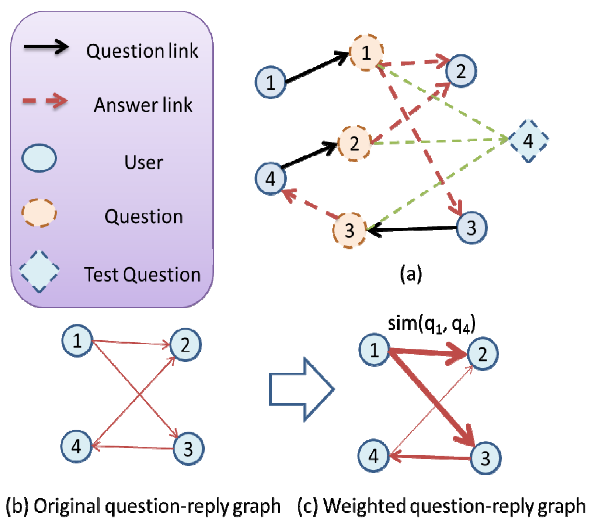

Figure 7: Demonstration of construction of weighted question-reply graph.

Before we use the Weighted PageRank algorithm, we construct the weighted question-reply graph in advance. Figure 7 shows an example of the construction of weighted question-reply graph. Figure 7 (b) is the original question-reply graph constructed by Figure 7 (a) and Figure 7 (c) is the weighted question-reply graph. We construct a weighted question-reply graph for each test question. Construction steps are described as follows. For a given test question q_4 in Figure 7, we calculate the edge strength between users using equation (8).

$$\boldsymbol{edge\_weight(q,u,v) = \sum_{v \leftarrow a_{kl},\ u \rightarrow q_k} sim(q, q_k)} \qquad (8)$$

where $edge\_weight(q,u,v)$ represents the edge weight from node u to node v with respect to question q. We use $u \rightarrow q_k$ to represent the user u posting the question q_k and $v \leftarrow a_{kl}$ to represent the user v answering the question q_k with the answer a_{kl}. The similarity measure we used is the cosine similarity technique as illustrated in equation (9).

$$\boldsymbol{sim(q_1,q_2) = \frac{\sum_{w \in q_1,q_2} f(q_1,w) * f(q_2,w) * (idf_w)^2}{\sqrt{\sum_{w \in q_1}(f(q_1,w) * idf_w)^2} * \sqrt{\sum_{w \in q_2}(f(q_2,w) * idf_w)^2}}} \qquad (9)$$

where $f(q,w)$ means the term frequency of term w in question q and idf_w means the inverse document frequency of the term w. Since the edge strength may be larger than 1, we normalize the edge strength by dividing the summation of all the edge strength using equation (10).

$$\boldsymbol{edge\_weight'(q,u,v) = \frac{edge\_weight(q,u,v)}{\sum_{u_1 \in V} \sum_{u_2 \in R(u_1)} edge\_weight(q,u_1,u_2)}} \qquad (10)$$

where $edge\_weight'(q,u,v)$ means the normalized value of $edge\_weight(q,u,v)$. After building the weighted question-reply graphs for every test question, we use the Weighted PageRank algorithm on these graphs. The original PageRank algorithm is defined by equation (11).

$$PR(u) = (1-d) * \frac{1}{N} + d \sum_{v \in I(u)} \frac{PR(v)}{|R(v)|} \qquad (11)$$

where $PR(u)$ is the authority value of the user u and $I(u)$ means the set of nodes pointing to u. The damping factor is represented as d and N means the total number of nodes in the graph. However, the test question is not considered in this formula. We utilized and augmented Weighted PageRank algorithm to calculate user's authority value to represent the expertise value of the user. The original PageRank formula is modified as

$$PeER(q,u) = (1-d) * \frac{1}{N} + d \sum_{v \in I(u)} PeER(q,v) * edge\_weight(q,v,u) \qquad (12)$$

where $R(u)$ means the set the nodes node which are pointed by node u. We set d=0.85 in our experiment. The final score of this algorithm assumes: if a user answered many questions similar to the test question asked by users with great authority value, he may be an authoritative user of this test question.

3.2 *Activity model*

As mentioned in the introduction section, we should route new questions to the users not only with great expertise but with great activity. User's activity score for a given question is composed of two components, *the period activity* and *the daily activity*. The former one estimates user's activity according to the answering curve of the user in the past, and we use the tendency of the curve to predict the activity value in the future. The latter one focuses on a more detailed information of time series. It makes use of the daily activity of users. We investigate user's active time per day. In other words, if one user always answers questions in the night, we should not route the questions posted in the morning to this user. Since questions cannot be solved efficiently, we should also consider the hour activity of users.

A real example of considering user's activity is illustrated in **Figure 8**. As shown in this figure, there are two users and their answering records with respect to time. One point represents one question is answered by the user at the time. In this case, we use the total answering count to represent the expertise score of these two users. User u_{9567} have answered 207 questions in the past and u_{78845} have answered 166 respectively. Thus the expertise of u_{9567} is higher than u_{78845}. Now we have a new question $q_{4572362}$ at time t_1. We route this new question to u_{9567} rather than u_{78845} according to the expertise score. However, we investigate the time series to these two users. We can easily get a conclusion that the activity of u_{78845} is higher than that of u_{9567} at time t_1 although u_{9567} have answered more questions. Therefore, we route the question $q_{4572362}$ to the user u_{78845} instead of the user u_{9567}. From this example, we consider user's activity is also an important criterion in question routing.

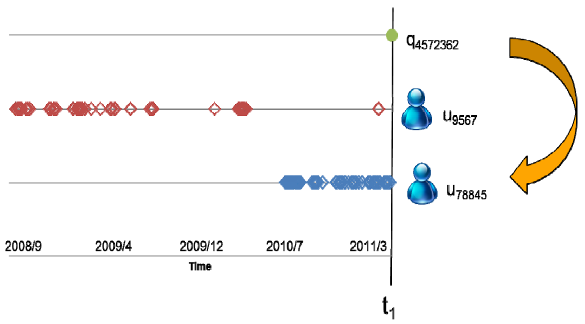

Figure 8: A real example of activity model.

PA: Period activity

In this sub-section, we describe the details of calculating the period activity. We consider the period activity of a user in two manners, i.e., query independent manner and query dependent manner. The former is denoted as PA^0 and the other is denoted as PA^q. Query independent means user's period activity is independent of the topic of the test question while question dependent assumes otherwise.

We predict user's activity in the testing region as the following steps. First, we do the discretization for the training region in advance. We assume that the length of the testing region is m days and the length of training region is n days. We split the training region into n/m bins with length m for each bin. In our experiment, we also set the length of bin as 4 days in PA^0 and PA^q since our testing region is 4 days. Since our problem is to solve the problem of question routing, we only focus on the questions users answered and use the answering count of each user to represent their activity value. After the steps of discretization, we can get the first seven values of each bin for every test user. We use SVM [Chang 2001], to predict the value and the kernel type we used is the sigmoid function. We train and predict user's activity by the regression model. We define this predicated score as PA^0. Users are ranked according to this score for each test question.

Since the method PA^0 does not consider the information of test question, we adjust the method PA^0 to develop another method PA^q. Instead of assuming each user having the same level of activity for different topics of questions, we consider question dependent to calculate user's activity. Users have different levels of activity for different topics of questions. This assumption makes sense since users usually have diverse background knowledge and experience.

When we count the answering questions for each bin, we only focus on the questions related to the test question. We calculate the similarity between the test question and the questions answered by candidate users. We scale the scale of similarity values into [0, 1].

DA: Daily activity

As the same definition of period activity, we also treat daily activity in two manners: DA^0 as the daily activity independent of the topic of test question and DA^q otherwise. Consider the example shown in **Figure 9**. The curve in the figure means the answering count versus hour of the user. In other words, a data point *(x, y)* means the user answered *y* questions in the hour *x*. we can see obvious difference between these two users in the figure. User u_1 is active in the night or morning while user u_2 is active at midday. Now a new question is posted at hour h_1. Assume these two users having the same expertise on this question. Since the activity of u_2 is higher than that of u_1 at hour h_1 according to the curves. We route the question q_1 to the user u_2.

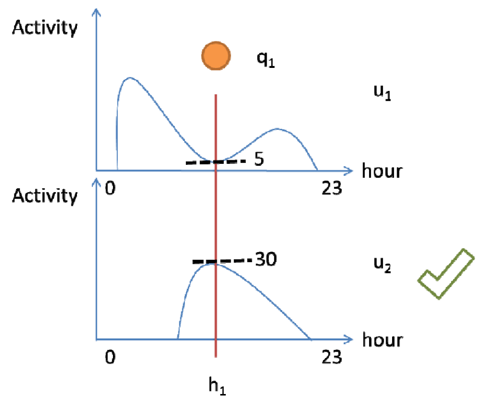

Figure 9: An example of considering hour activity.

In order to develop our method, DA^0 and DA^q, we collect data from 2010/10/28 ~ 2010/10/31 from the website, Stack Overflow, to investigate the response time of each question. There are 6,160 questions and 11,272 answers during this period. Most answer's response time is usually short. On average, the time interval of answer and question is 2.26 hours.

We calculate DA^0 and DA^q as follows: For a given question q, supposed q is posted at hour h_1, we calculate the score of the user's hour activity score by counting the answering questions the user answered from hour h_1 to h_1+*average response time* in the past as DA^0. *Average response time* means the average response time of all answers of all users in our training set. We also consider questions dependent with the hour activity. We only count the relevant questions to the test question to be the score DA^q.

We then use a linear regression model to combine the expertise model and activity model. Hence, we rank users for each test question by the combined score $Expertivity(q, u)$.

4. Experiments

In this section, we present experimental evaluation results to access the effectiveness of our system. In particular, we conduct experiments on the Stack Overflow (http://stackoverflow.com) cQA archive with over tens of thousands questions in our work. This dataset is public and can be downloaded from the Stack Overflow Blog (http://blog.stackoverflow.com/). The dataset we downloaded covers the duration from July 31, 2008 to December 31, 2010 and what we used for our experiment is the snapshot from October 1, 2010 to October 31, 2010. These questions are limited in the "programming" domain. The dataset statistics are shown in **Table 2**. There are totally 71,000 questions and

123,739 answers which belong to these questions during the duration. There are total 51,888 users participating in these questions and answers. Among these users, we separate askers and answerers in order to investigate the ratio they overlap. This results in 33,664 askers and 29,482 answerers. Furthermore, we remove the askers from answerers to count number of answerers only and number of askers only respectively. There are 18,224 (61.8%) users who answer questions only without asking questions among the answerers and 22,406 (66.6%) users who ask questions only without having answers to others' questions among the askers respectively. Among these users, the proportion of users who both have questions and answers is 11,258 (21.7%). Due to this characteristic, the question-reply graph tends to the bipartite graph which means that majority of users who have answers to others' questions have no questions in the cQA forum.

Table 2 : Dataset statistics

Num of questions	71,000
Num of answers	123,739
Num of users	51,888
Num of askers	33,664
Num of answerers	29,482
Num of answerer only	18,224
Num of asker only	22,406
Num of both answerer-askers	11,258
Average number of answers per question	1.83
Max number of answers per question	39
% answerers with 1 answer	16,753

Considering our problem in this paper, we want to route a new question to the right users to get answers with good quality efficiently. Therefore we need to use the information (i.e. asking questions and answering questions) of each user. To obtain the judgment whether a new question suits to a user or not, we separate our data into two parts, *Set A* and *Set B*. *Set A* covers the duration from October 1, 2010 to October 27, 2010 and the rest for *Set B*. We use *Set A* for training and *Set B* for testing respectively. First, we extract questions from *Set B*. For each question, we filter out the questions with no answers. In next step, we eliminate the questions whose answerer with no answering history in *Set A*. As a result, we obtain 6,160 test questions. The test users are collected from the answerers answering test questions. This results in 3,492 test users to be ranked. We use all answerers of each test question as ground truth 1 (denoted as GD_1). For each test user, we use the questions in their answering history in set A for training. On average, each test user has answered 17 questions while the max is 596 and the min is 1.

In order to evaluate the performance of our system and to compare with other methods, we use four evaluation metrics including MAP, MRR, R-Precision, Precision@k to calculate the performance score. MAP is the abbreviation of Mean Average Precision. It represents the mean of the average precisions (denoted as AP) over the set of query questions. In our work, the query questions represent the new questions to be routed. Average precision can be defined as the formula:

$$\boldsymbol{AP} = \frac{\sum_i \boldsymbol{P@i} * \boldsymbol{corr(i)}}{\boldsymbol{No. of\ users\ answering\ the\ test\ question}} \quad (13)$$

where i is the rank, and $corr()$ is a binary function on the relevance of a given rank, and $P@i$ is the precision at a given cut-off rank.

$$\boldsymbol{MAP} = \frac{\sum_{q=1}^{Q} \boldsymbol{AP(q)}}{|\boldsymbol{Q}|} \quad (14)$$

where Q is the set of query questions. The mean reciprocal rank (MRR) [Voorhees 1999] is the average of the reciprocal ranks of the first correct answers over a set of query questions Q. It is defined as follows:

$$\boldsymbol{MRR} = \frac{\boldsymbol{1}}{|\boldsymbol{Q}|}\sum_{i=1}^{U} \frac{\boldsymbol{1}}{\boldsymbol{rank(i)}} \quad (15)$$

where $rank(i)$ is position of first recommended user retrieved. MAP considers all answers while MRR only considers the first answer. We use this metric as our major evaluation measure, because we want to know how many users we should route to get at least one answer. R-Precision is the precision at rank *R* where *R* is the total number of answerers for each test question in ground truth. We also retrieve P@1 and P@3 to calculate the percentage of *top-k* candidate users retrieved that are correct. We utilized the following baseline methods to compare with our system in order to demonstrate the effectiveness of our approach:

Reply Count (RC): This method use the number of questions replied by the user as the user's score.

PageRank (PR): We apply PageRank algorithm on the question-reply graph to rank users by their authority values [Zhang 2007]. The question-reply graph is constructed from *Set A* dataset. We remove the self-cycle which indicates one user answers the question asked by him/her in the graph. As a result, there are total 36,799 nodes and 103,274 edges in the graph. We use this method to estimates the global authority score of each user since a user with great authority has more probability to answer new questions.

Average Response Time (ART): This method estimates the score of each user by calculating the average response time between their answers and corresponding questions. We define this as follows:

$$score(u) = \frac{\sum_{a \in A_u}\left(time(a) - time\big(q(a)\big)\right)}{|A_u|} \quad (16)$$

where A_u represents the set of answers provided by user u. $time(a)$ means of creation time of the answer a while the creation time of the corresponding question of the answer a is represented as $time\big(q(a)\big)$. This baseline is used to compare with our activity model since we assume that the average response time of users and user's activity are positively correlated.

Entropy model: We estimate the score of each candidate user by entropy calculation. The formula is defined as follows:

$$score(u) = -\sum_{i=1}^{n} p(x_i) log\big(p(x_i)\big) \quad (17)$$

where n means the total number of bins and i means the *i-th* bin as we defined in the activity model and $p(x_i)$ means the number of questions answered by the user in the bin dividing his total answering questions.

Inverse Entropy model: We also use this model to be the baseline of our activity model. The score is calculated by inverse the score of Entropy model.

Table 3 shows the performance of our system and comparison with other methods. In the expertise model, PeER is better than QLL. The findings indicate that peer-expertise model is more effective than content-based method. Graph-based method is more effective than content-based method in determining the expertise of users. According to the result of combination of PeER and QLL, we can get a better performance than using these two methods separately since the MRR value of QLL is 0.0999 and that of PeER is 0.1222 while that of combination of these two methods is 0.1295.

In our activity model, we examine of effect of question dependent in advance. PA^q is better than PA^0 and DA^q is better than DA^0. The results revealed that question dependent manner is important than question independent. We should consider the information when we calculate the activity of users. PA^q is the most effective method. It uses the period activity of users by modeling the answering curve according to time. DA^q is slightly worse than PA^q. Since PA^q do not consider the user's active time per day, we combine this method with DA^q. We combine these two methods to enhance the activity model. The MRR value of activity model is 0.1245, while that of PA^q is 0.1176 and that of DA^q is 0.1023 respectively.

We use three methods to compare with our activity model, namely Average Response time, Entropy, and Inverse Entropy. We compare these three baselines with PA^q. We can see from Table 3 that PA^q outperforms the other three baselines no matter what evaluation metric is used. The performance of Entropy outperforms Inverse Entropy. The results revealed that there are few phenomenon of busty event in user's answering record versus time.

Despite considering the test questions, we compare the methods RC, PR, PA^0. Among these methods, the performance of PA^0 is the best. We compare with RC and PA^0. RC uses the answering count scores to rank users while PA^0 models the answering count versus time. We can get a conclusion that some users are not always active on the website. We cannot use the total answering count to determine the score of users. We should consider the trend in user's answering curve instead of just counting the total number of answering questions.

As mentioned in introduction, we should consider the user's expertise and activity simultaneously in the problem of question routing. In the combination part, expertise model is more effective than activity model. We compare our expertise model and activity model. The performance of expertise model outperforms the activity model. If we want to route a new question to users, user's expertise is more important than user's activity. Moreover, according to the result of combination of expertise model and activity model, we get the best system performance when the weights of these two models equal 0.5. These two models are equivalently important in our problem. Our final system performance yields the best performance comparing the other methods. The best MRR value is 0.1372.

Table 3 : Overall comparison of all methods in *GD_1*

Method	*MAP*	*MRR*	*R-Precision*	*P@1*	*P@3*
RC	0.0219	0.0331	0.0108	0.0122	0.0094
PR	0.0166	0.0252	0.0059	0.0081	0.0053
ART	0.0018	0.0026	0.0003	0.0003	0.0002
Entropy	0.0160	0.0236	0.0078	0.0088	0.0056
Inverse Entropy	0.0018	0.0024	0.0004	0.0002	0.0003
QLL	0.0731	0.0999	0.0457	0.0497	0.037
PeER	0.0891	0.1222	0.0633	0.0732	0.0449
PA^0	0.0223	0.0333	0.0107	0.0122	0.0096
PA^q	0.0828	0.1176	0.0582	0.0708	0.0426
DA^0	0.0223	0.034	0.0108	0.0101	0.0108
DA^q	0.0706	0.1023	0.0494	0.0596	0.0361
Expertise Model	0.0955	0.1295	0.0693	0.0810	0.0463
Activity Model	0.0874	0.1245	0.0629	0.0755	0.0451
Expertivity	**0.0997**	**0.1372**	**0.0734**	**0.0857**	**0.0494**

5. Conclusion

In this paper, we address the problem of question routing over a community QA portal, Stack Overflow. We show that not only user's expertise but user's activity is also an important criteria for question routing. Based on our proposed framework, we have introduced several methods to model user's expertise and activity. Expertise model includes QLL and PeER; QLL uses content-based method while PeER uses peer-expertise model. Activity model contains PA^q and DA^q; PA^q uses the answering curve of each user while DA^q uses daily active time of users to model user's activity.

Our experiments on a collection of Stack Overflow cQA portal have shown that our proposed methods outperform the baseline methods and enjoy a better performance. Our proposed method has the best performance in several measurements. Moreover, our method combined by expertise model and activity model is also better than the method using training technique. The best MRR value in our experiments is 0.1367. It means that on average each test question will get at least one answer if we route the test question to the top 7 ranked users. Considering the total users to be ranked, our system demonstrates that our proposed framework is able to route new questions to users. New questions can be solved effectively and efficiently.

References

[Adamic 2008] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and yahoo answers: everyone knows something," presented at the Proceeding of the 17th international conference on World Wide Web, Beijing, China, 2008.

[Agichtein 2008] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," presented at the Proceedings of the international conference on Web search and web data mining, Palo Alto, California, USA, 2008.

[Balog 2006] K. Balog, L. Azzopardi, and M. d. Rijke, "Formal models for expert finding in enterprise corpora," presented at the Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA, 2006.

[Brin 1998] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," presented at the Proceedings of the seventh international conference on World Wide Web 7, Brisbane, Australia, 1998.

[Campbell 2003] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom, "Expertise identification using email communications," presented at the Proceedings of the twelfth international conference on Information and knowledge management, New Orleans, LA, USA, 2003.

[Cao 2008] Y. Cao, H. Duan, C.-Y. Lin, Y. Yu, and H.-W. Hon, "Recommending questions using the mdl-based tree cut model," presented at the Proceeding of the 17th international conference on World Wide Web, Beijing, China, 2008.

[Chang 2001] C.-C. Chang and C.-J. Lin, "LIBSVM : a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm," 2001.

[Craswell 2005] N. Craswell, A. P. d. Vries, and I. Soboroff, "Overview of the TREC-2005 Enterprise Track," presented at the In Proceedings of TREC 2005, 2005.

[Dom 2003] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang, "Graph-based ranking algorithms for e-mail expertise analysis," presented at the Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, San Diego, California, 2003.

[Gyongyi 2007] Z. Gyongyi, G. Koutrika, J. Pedersen, and H. Garcia-Molina, "Questioning Yahoo! Answers," Stanford InfoLab, Technical Report2007.

[Jeon 2005] J. Jeon, W. B. Croft, and J. H. Lee, "Finding semantically similar questions based on their answers," presented at the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil, 2005.

[Jeon 2006] J. Jeon, W. B. Croft, J. H. Lee, and S. Park, "A framework to predict the quality of answers with non-textual features," presented at the Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA, 2006.

[Jurczyk 2007] P. Jurczyk and E. Agichtein, "Discovering authorities in question answer communities by using link analysis," presented at the Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, Lisbon, Portugal, 2007.

[Kleinberg 1999] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM,* vol. 46, pp. 604-632, 1999.

[Li 2010] B. Li and I. King, "Routing questions to appropriate answerers in community question answering services," presented at the Proceedings of the 19th ACM international conference on Information and knowledge management, Toronto, ON, Canada, 2010.

[Littlepage 1997] G. E. Littlepage and A. L. Mueller, "Recognition and Utilization of Expertise in Problem-Solving Groups: Expert Characteristics and Behavior," *Group Dynamics: Theory, Research, and Practice,* pp. 324-328, 1997.

[Liu 2008] Y. Liu, J. Bian, and E. Agichtein, "Predicting information seeker satisfaction in community question answering," presented at the Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, Singapore, Singapore, 2008.

[Ponte 1998] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," presented at the Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, 1998.

[Qu 2009] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen, "Probabilistic question recommendation for question answering communities," presented at the Proceedings of the 18th international conference on World wide web, Madrid, Spain, 2009.

[Suryanto 2009] M. A. Suryanto, E. P. Lim, A. Sun, and R. H. L. Chiang, "Quality-aware collaborative question answering: methods and evaluation," presented at the Proceedings of the Second ACM International Conference on Web Search and Data Mining, Barcelona, Spain, 2009.

[Voorhees 1999] E. M. Voorhees, "The TREC-8 question answering track evaluation," *TREC,* pp. 83-105, 1999.

[Wang 2009] K. Wang, Z. Ming, and T.-S. Chua, "A syntactic tree matching approach to finding similar questions in community-based qa services," presented at the Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, Boston, MA, USA, 2009.

[Wu 2008] H. Wu, Y. Wang, and X. Cheng, "Incremental probabilistic latent semantic analysis for automatic question recommendation," presented at the Proceedings of the 2008 ACM conference on Recommender systems, Lausanne, Switzerland, 2008.

[Xing 2004] W. Xing and A. Ghorbani, "Weighted PageRank Algorithm," presented at the Proceedings of the Second Annual Conference on Communication Networks and Services Research, 2004.

[Zhai 2004] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Trans. Inf. Syst.,* vol. 22, pp. 179-214, 2004.

[Zhang 2007] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," presented at the Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, 2007.

[Zhou 2009] Y. Zhou, G. Cong, B. Cui, C. S. Jensen, and J. Yao, "Routing Questions to the Right Users in Online Communities," *Proceedings of the 2009 IEEE International Conference on Data Engineering,* pp. 700-711, 2009.

APPENDIX

Table 4 : Contents of Answering questions of User u_{405015} in Stack Overflow dataset

Question id	*Content of question*

$q_{6618485}$

On my sandbox site there is a white bar across the top. It is contained within the body(which has the background image), and the navbars styles are

```
#topnav4 {background: #fff;  width: 100%; height: 70px;  }
```

But the bar doesn't reach the edges of the screen. Furthermore, I don`t have any margins or padding set. does anyone know how to fix this?

Update, also note that the navbar is not at the very top of the screen. There is space between the top of the navbar and the top of the page. How do I get rid of this?

Update, I added this code but the problem is still not fixed

#topnav4 {background: #fff; position: absolute; left: 0px; top: 0px; width: 100%; height: 70px; }

$q_{6614999}$

This is strange. I'm trying to have a fixed-width div next to a right-floated div, and I don't want to reorder the divs (because this is distributed theme). So I'm playing with negative margin-right on the fixed div, and I get what seems to me strange: if it's -4px or less, then the float moves to the side; otherwise, it stays below.

Play with the live demo with code at jsbin, which has this:

```
<style>
  .container {
    width: 200px;
    height: 200px;
  }
  .box {
    width: 100px;
    height: 100px;
  }
  .one {
    margin-right: -4px; /* If <= -4, .two box shifts up */
    display: inline-block;
  }
  .two {
    float: right;
  }
</style>
  <div class="container">
    <div class="box one"></div>
    <div class="box two"></div>
  </div>
```

Can someone explain the mystery? What's special about the number -4 in this case?

$q_{6614886}$

I'm wondering if that (my title) is ever incorrect, other than for HTML validation. I've recently had to start supporting IE7 again (I've been lucky enough to not have to for the past 3 years or so) and the fact that div's can't be `inline-block` has gotten me about 10 times in the past month due to the fact that I make everything a `div` by default and then go back and stylize elements. So I'm considering making ***everything*** a span so that if I later go back and make something inline-block I'm not trying to figure out why it's not working in IE7.

So my question -- Is there ever a case, in any browser (IE7+, FF, Webkit, Opera), that anyone knows of where a span can not act like a div? I'm not concerned about the HTML not validating due to having block elements inside inline ones.

$q_{6600523}$

I seem to have a glitch on in Internet Explorer and wondered if anyone could shed some light

Taking the following page as an example

http://www.flipfilter.com/websites-for-sale

When running in IE not under compatibility mode, all the Cufon (js font replacement) headings disappear. When I click the icon to enable compatibility mode, the pagination seems to go crazy and extends down the page.

Can anyone point me in the right direction, specifically as to what causes either of these problems?

(If the problem is a validation issue could you point me to which one specifically causes the two errors?

Many Thanks

$q_{6598083}$

I have a script in the link below which works in IE8 nicely. However, it seems to go wrong a little in IE7.

In IE7, I end up getting a bottom scroll bar with the text in the 2nd column behind the vertical scrollbar.

The text is supposed to move a little automatically to the left if a scrollbar is added because of the `overflow:auto` in the css.

How can I get this to work so it resizes properly without having the text go behind the vertical scrollbar?

Here is a jsfiddle of it in action: http://jsfiddle.net/9vEdw/