Questionnaire generation for better opinion polling

submitted for blind review

No Institute Given

Abstract. This paper studies a novel problem. Given a topic, how to automatically generate a polling questionnaire from a question bank collected from online debate forums? This is not a trivial task, as even opinion polls composed by human experts are not always reliable. We propose a framework which is based on a bi-level topic sensitive question graph, and model the questionnaire problem as selecting a minimal independent set that covers a maximal weighted set of key phrases. We further present statistical technologies to address the challenges of matching questions that are both semantically similar and producing similar respondences. Both quantitive and qualitative experimental results demonstrate the competency of our methods.

1 Introduction

Opinion poll is not uncommon in modern society. The first known example of an opinion poll was conducted nearly 200 years ago[1]. It successfully predicted the outcome of United States presidency in 1824. Since then, it has been a convention to organize an opinion poll to measure public opinions and document the experiences of the public. Nowadays, opinion polls cover a wide range of subjects, from market performance of a video game to presidential job approval ratings. Polling results provide valuable information for everyone on events in the news and other topics of interest, and facilitate in policy making and other decision making.

In an opinion poll, a series of questions are handed out to voters sampled from the target population. The goal of the poll is to extrapolate generalities based on the answers by sample voters. Thus the design of questionnaire is crucial. Conventionally polling questions are composed by professionals with expertise. There are polling organizations in virtually every country with elections. Major television networks also operate opinion polling regularly. However, opinion polls are not always reliable in predicting public opinions on an unseen topic. For instance, Donald Trump's election in 2016 signals the biggest and complete poll failure ever.

Recently, online social media sites has become a dominant platform for online users to public and exchange ideas and opinions. It has attracted researchers to study public opinions [11,6]. Online debate forums is in particular valuable for

[1] `https://en.wikipedia.org/wiki/Opinion_poll`

public opinion mining. In a debate forum, many simple agree/disagree scale questions are presented, and users are invited to vote and comment on one option. Some example questions and their comments are shown in Fig. 1 .

From the view of opinion polling, a debate forum is a bank of polling questions. Is it possible to automatically generate an opinion poll for any given topic from online debate forums? In this paper, we explore the possibility of generating opinion polls more rapidly and less expensively. The following scenario is presented, given a crawl of online debate forums, a set of topics, the task is to automatically select appropriate polling questions from the crawl for any topic.

The task of question selection is far from trivial. There are some general guidelines to write questions in the community of pollsters. For example, it is well established that a poll should ask "enough questions to allow all aspects of an issue to be covered". The wording of each question should be carefully organized so that the true feelings of respondents are revealed [2].. Keeping the questionnaire as short as possible is also a golden role [14]. However, there is no known computational solution.

We propose a framework which is based on a **Bi**-level **T**opic **S**ensitive **Q**uestion **G**raph (BITSQG). The first layer of bi-level topic sensitive question graph represents the equivalences between questions, the second layer consists of key phrases weighted by how relevant they are to the given topic. To produce a polling questionnaire, we extract the minimal independent set of questions covering a maximal weighted set of key phrases.

There are two challenges raised in the graph based framework. (1) How to match equivalent questions? Determining equivalent polling questions is beyond paring up semantically similar questions. Small changes of words in a question can lead to big changes in the answer. For example, in the toy example in Fig. 1, it is difficult for teens to "obtain" marijuana, but it is easy for teens to "buy" marijuana[3].(2) How to extract key phrases and associate a topic relevance with each key phrase? On one hand, the questions are too short to provide enough co-occurrence information for any significance measure. On the other hand, questions are usually well formatted and the internal structure might be useful in detecting key phrases.

Our contributions in this paper are three fold. (1) We study the problem of opinion poll generation for a given topic from a question bank collected from online debate forums. To the best of our knowledge, this is a novel problem. To tackle this problem, we present a framework which is based on a bi-level topic sensitive question graph. (2) We combine semantic similarity and response similarity to match equivalent questions. We utilize the statistical concept margin of error to detect questions with distinguishing responses. (3) We employ sequential pattern mining techniques in both questions and comments to extract and weigh key phrases. We conduct quantitive and qualitative experiments on real data sets to demonstrate the competence of our proposed methods.

[2] `https://en.wikipedia.org/wiki/Opinion_poll`

[3] The toy example is a modified version of 2003 survey of teens and drug uses

This paper is structured as follows. We describe details of our framework and algorithms in Sec 2 and Sec 3 . The experimental results are presented and analyzed in Sec 4. We briefly introduce related work in Sec 5. We give our introduction and a outlook for future work in Sec. 6.

q_1: Is it difficult for someone your age to buy marijuana?	
Pro: 390	Con: 610
Lexy: We still need doctor's documents	Alice: A lot of drug selling occurs in my neighborhood
...	...
q_2: Is it easy for someone your age to obtain marijuana?	
Pro: 150	Con: 350
Alice: In my experience, fairly simple	Bob: Marijuana seeds are not everywhere
...	...
q_3: Is it easy for someone your age to find marijuana?	
Pro: 630	Con: 270
Shania: Most of my friends used marijuana, if not all	Coco: Not in medical marijuana clinics
...	...
q_4: Should governments legalize marijuana?	
Pro: 590	Con: 210
Fanny: It should be legalized, regulated, and taxed.	Anon:No it makes people lazy and unproductive. Driving while high is a serious problem.
...	...
q_5: Should marijuana be illegal?	
Pro: 185	Con: 420
Tygaa: Marijuana makes you have a slow reaction to every thing. This drug makes you have car crashes.	Arget: It should be taxed and regulated. Marijuana is less harmful than tobacco and alcohol.
...	...

Fig. 1. An illustrative example of opinion poll generation

2 Question Graph

The problem setup is based on online debate forums. An online debate forum as a set of questions $Q = \{q\}$, Each question is a tuple $q = (T_q, N_q, P_q)$, where T_q is the question's textual content. We only deal with dual-sided debates, thus P_q is the pro-stance, and N_q is the con- stances. In online debate forums, users argue why their viewpoint is right by commenting on the chosen vote option. We model N_q, P_q as a set of vote-comment pairs $< \boldsymbol{u}, c >$, where $\boldsymbol{u}$ is the user portfolio vector, and c is the comment's textual content. We adopt the bag-of-word representation for all text segments T_q, c.

To enable the application of graph-based opinion poll generation, we first introduce the concept of question graph. A question graph is a undirected graph $G = (Q, E)$ formed by a set of nodes and a set of edges. Without ambiguity, we use the same symbol q to denote both questions and nodes in the question graph. Each edge $e(q, q') \in E, q, q' \in Q$ connecting q and q' indicates that the corresponding questions q, q' are interchangeable. For the cause of opinion polling, we make the hypothesis that, two questions q, q' are interchangeable if (1) they are semantically similar (2) they yield similar polling results. Therefore we first present a clustering algorithm to construct cliques of semantically similar questions, and then we break ties between questions that result in distinguishing estimated responses.

2.1 Semantic Clustering

As we can not specify the number of semantic clusters, we need a non-parametric clustering algorithm. For this purpose, we use the **A**ffinity **P**ropagation (AP) algorithm. The idea of AP is to identify exemplars and assign remaining data points to nearest exemplars. The algorithm stores information in two matrices: R, A. The "responsibility" matrix $R = \{r(i, j)\}$ measures how well a data point j can represent i. The "availability" matrix $A = \{a(i, j)\}$ quantifies how appropriate it is for i to pick j as an exemplar.

The AP algorithm starts by initializing both matrices to all zeroes, and proceeds to update R, A iteratively. R is updated by passing messages from cluster members to cluster exemplars. A is updated by passing messages from cluster exemplars to cluster members.

$$r(i, j) \leftarrow s(i, j) - \max_{j' \neq j} \{a(i, j') + s(i, j')\} \tag{1}$$

$$a(i, i) \leftarrow \sum_{i' \neq i} \max(0, r(i', i)) \tag{2}$$

$$a(i, i) \leftarrow \sum_{i' \neq i} \max(0, r(i', i)) \tag{3}$$

The updating messages are based on similarity function $s(i, j)$. It is worth noted that affinity propagation doesn't require the similarities to be symmetric, as long as the similarity function satisfies the following properties.

Property 1 $s(i, j) > s(i, j')$ if and only if i is more similar to j than to j'.

Since the similarity function plays an important role in AP, we tailor a similarity measurement for question clustering. Intuitively, key phrases are most significant featurs of questions. We use $SF(q) = \{k, k \in T_q\}$ to denote the key phrases contained in question q. The extraction of key phrases will be introduced in Sec. 3. The similarity function is defined as:

$$s(i, j) = \frac{|SF(i) \bigcap SF(j)|}{|T_i|} \tag{4}$$

According to Equ. 4, if the two questions have more common key phrases, then they are more similar. The dominator is only related to question i. If two pairs $< i, j >$ and $< i, j' >$ have the same number of common key phrases, then the similarities $s(i, j) = s(i, j')$. This ensures that the length of questions as cluster members will not affect the "responsibility" and "availability" of the exemplars.

The AP algorithm stops when the cluster boundaries stay unchanged or the maximal number of iterations is achieved. The nodes with positive self similarities are chosen as exemplars.

2.2 Questions with Distinguishing Responses

We have shown in Sec. 1 that, even slight wording differences between semantically similar questions can lead to distinguishing responses. To generate an accurate polling questionnaire, it would be ideal to measure question similarities in the sense of wording quality. However, the effects of wording are too complicated to derive a simple similarity metric. We present to detect questions with dissimilar wording choices by directly using the responses.

In online debate forums, each question is voted and commented by a group of users. For simplicity, we presume that this group of users get together randomly. The question-answering process is equivalent to conducting a polling with one question. In statistics, the margin of error is used to interpret the polling result. For a question q, the ratio of people holding a pro-stance is $\sigma(q) = |P_q|/(|N_q \bigcup N_q|)$, the margin of error is $MOE(q) = \frac{1}{\sqrt{n(q)}} \times 100\%$, where $n(q) = |P_q \bigcup N_q|$ is the number of votes (and comments) for question q, then the estimated response interval will be:

$$I(q) = [\sigma(q) - MOE(q), \sigma(q) + MOE(q)] \tag{5}$$

For example, 1000 people answered question q_1 in Fig. **??**, 61% of them think that buying marijuana is easy. Based on this sample, we are 95% confident that between 58% and 64% of all users on the debate forum think that buying marijuana is easy. 500 people answered question q_2, 30% of them consider obtaining marijuana is easy. We can predict that the percentage of people who agree that obtaining marijuana is easy will be in the range of $[25.5\%, 34.5\%]$ with 95% confidence.

Intuitively, if two questions are both worded nicely to reflect the true public opinion, then the resulting output must accord with each other. If the two questions have non-overlapping response ranges, then the wordings must be distinguishing. Based on this hypothesis, we present the following steps.

After semantic clustering in Sec 2.1, we construct cliques of queries, where every pair of queries are connected. Then we remove edge $e(q, q')$ between questions who have non-overlapping response intervals $I(q) \bigcap I(q') = \emptyset$. For example, for the questions in Fig. 1, we first construct two cliques, then we remove the edges between q_1, q_2 and q_2, q_3. The steps are shown in Fig. 2.

Fig. 2. Constructing a question graph in the toy example

We measure the response similarity based on estimated portion of pro-stances. There is one more issue left: how to match the pro-stances between a pair of questions? The pro-stances for each question do not always refer to the same opinion. For example, the pro-stance of question q_1 is actually mapped to the con-stance of question q_2 in Fig. 1.

We treat the pro-stance matching task as a binary classification task. Suppose we are given two questions q, q', we train a classifier so that the output is positive if P_q matches to $P_{q'}$ while output is negative if P_q matches to $N_{q'}$. We use features from 4 fields: text, vote, demographic and sentiment. From each field, we derive two features indicating how much the output will be positive and negative respectively.

As shown in the toy example, the stance can be recognized from the comments. For example, the pro-stance of q_3 and the con-stance of q_4 are much alike in vocabulary. The text field is the combined textual body of the question and all of its comments. Intuitively, two stances are identical if their text fields are similar. We use cosine similarity to compare two stances. f_1 is a feature correlated to positive output, f_2 is correlated to negative output.

$$f_1 = cosine(P_q, P_{q'}) + cosine(N_q, N_{q'}) \tag{6}$$
$$f_2 = cosine(P_q, N_{q'}) + cosine(N_q, P_{q'}) \tag{7}$$

The stance can also be identified if a user votes for both questions. Hypothetically, a user will stick to one stance. For example, in the toy example, Alice votes for both q_1, q_2, and the con-stance in q_1 is the pro-stance in q_2. For a pair of instances, the more common voters they have, the more likely the two stances will be matched.

$$f_3 = |\{u|u \in P_q, u \in P_{q'}\}| + |\{u|u \in N_q, u \in N_{q'}\}| \tag{8}$$
$$f_4 = |\{u|u \in P_q, u \in N_{q'}\}| + |\{u|u \in N_q, u \in P_{q'}\}| \tag{9}$$

We further extend the hypothesis on individual vote to group votes. We believe that for a group of users with similar demographic attributes, the opinion distribution will be stable. We consider groups determined by party (demographic and republicans) and age (20−?20 − 40?40+). For each group g on

question q, we compute the pro-stance frequency $ps(q,g) = [|\{u : u \in g, u \in P_q\}|$ and the con-stance frequency $cs(q,g) = |\{u : u \in g, u \in N_q\}|]$. We summate cosine similarities between stance frequency vectors over all groups.

$$f_5 = \Sigma_g cosine([ps(q,g), cs(q,g)], [ps(q',g), cs(q',g)]) \tag{10}$$
$$f_6 = \Sigma_g cosine([ps(q,g), cs(q,g)], [cs(q',g), ps(q',g)]) \tag{11}$$

We exploit the sentiments in the query to match stances. We extract negative opinion words by the FBS lexicon [5]. Suppose $n(q)$ is the number of negative opinion words in question q, since a double negative turns the opinion to a positive, we have

$$f_7 = 1 - [n(q) - n(q')]\%2 \tag{12}$$
$$f_8 = [n(q) - n(q')]\%2 \tag{13}$$

3 Questionnaire Generation

3.1 Bi-level Topic Sensitive Question Graph

The question graph constructed in Sec. 2 is static. To generate the polling questionnaire for a given topic, we first need to build a dynamic graph induced by the topic. A BITSQG induced by topic o consists of two layers $BITSQG^o = (Q, E, K, W^o, A)$. The first layer is the question graph, the second layer is a set K of key phrases, where each node is assigned a weight $w^o(k), k \in K$. The two layers are connected by arcs in A. An arc $a(q,k)$ is added to the BITSQG if the question q (either in the question body or in one of the comments) contains key phrase k. An illustrative BITSQG is shown in Fig. 3

We extract key phrases in questions as the basic semantic units. Conventionally, key phrase extraction is implemented by filtering out statistical significant phrases. However, the questions are too short and sparse and do not offer enough statistics. We observe that questions in online debate forums are well formatted. For example, many questions are in the form "should ... be ...?" We employ sequential pattern mining algorithm to detect common question templates.

For a question which satisfies several templates, we remove the longest template. The remaining part is considered to be a key phrase. For example, question "Should capital punishment be legal ?" satisfies two templates "should...be..." and " should... be legal". We delete the longest template from the question and obtain capital punishment.

The weights of key phrases in BITSQG indicate the relevance of key phrases to the topic. The relevance depends on co-occurrence in both question and comments. The relevance $w^o(k)$ of each key phrase k to a given topic o is defined by

$$w^o(k) = w_1 \times |Q(o) \bigcap Q(k)| + w_2 \times (|Q(o) \bigcap C(k)| + |Q(k) \bigcap C(o)|) \tag{14}$$

where $Q(o)$ is the set of queries containing the key phrase o in the query texts, $C(k)$ is the set of queries containing the key phrase k in the comments, w_1, w_2 are combinational weights.

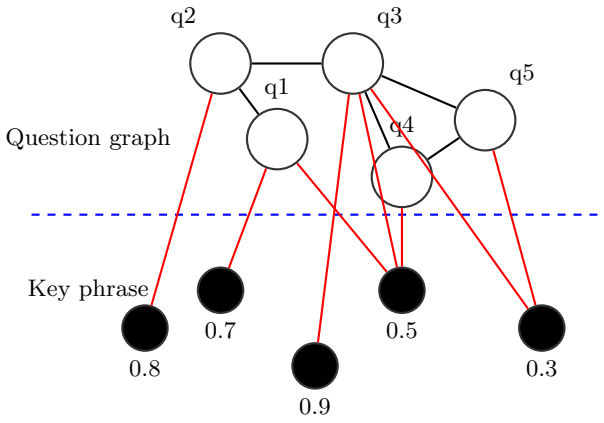

Fig. 3. An illustration of bi-level topic sensitive question graph

3.2 Minimal Independent Questions Covering Maximal Key Phrases

The questionnaire needs to be concise, while preserving the coverage of relevant key phrases. In BITSQG, two questions are connected if they are semantically while and they produce similar polling responses. This means that if we have two adjacent questions in the result, the questionnaire will be redundant. In other words, our target is an independent question set. An independent question set $S \subset Q$ is a set of questions where no two nodes are adjacent.

For an arbitrary set $S \subset Q$ of questions, the sum of weights of key phrases it covers on topic o is defined as

$$w^o(S) = \Sigma_{q \in S, a(q,k)} w(k)^o \tag{15}$$

To generate the questionnaire, we aim to select a **M**inimal **I**ndependent set covering a **M**aximal **W**eighted set of key phrases (MIMW). S is a MIMW that (1) S is an independent question set; (2) there is no independent set S' that $W^o(S') > W^o(S)$; (3) there is no independent S' with $W^o(S') = W^o(S)$ that $|S'| > |S|$. For example?in Fig. 3, the minimal set covering the maximal weighted set of key phrases is $\{q_2, q_3\}$.

The MIMW problem is NP-hard given that the independent set problem is known to be NP-hard. A greedy algorithm approach can achieve approximate rate of $1 + \log(\Delta \| OPT \|)$, where Δ is the maximal degree of $BITSQG$ and

OPT is the optimal set. We present the greedy algorithm to select a subset of questions. Starting from an empty set, at each step, we select the question whose covering weight is maximum .

Algorithm 1: Greedy $MIMW$ Approximation

Input: $BITSQG = (Q, E, K, W^o, A)$
Output: MIMW set S
$S \leftarrow \emptyset$;
while $|K| \neq \emptyset$ **do**
  for $q \in Q$ **do**
    $w(q) = \Sigma_{k \in K, a(q,k)} w(k)^o$;
  end
  $q* = \arg\max_{q \in Q} w(q)$;
  $S \leftarrow S \bigcup \{q*\}$;
  $Q \leftarrow Q - \{q*\}$;
  $K \leftarrow K - \{k | a(q*, k) \in A\}$;
end

4 Experimental Analysis

4.1 Experimental Setup

The data set used in the experiment is obtained from a large online debate forum[4]. We crawl 14,620 questions published before February 2nd, 2017 on the largest topic: politics. For each question, we crawl all the votes and comments (152946 votes), We crawl the user profiles for each user participating in at least one question. The user demographics include party, ethnicity, age group and other 22 attributes.

In pre-processing, we remove anonymous users and questions without comments, reduce the number of questions to 6777 and the number of comments to 28,911. We remove english stop-words and run Porter Stemmer. Stop-word removal and stemming significantly decrease the length of questions and comments. The default weighing scheme is tf-idf. The statistics is shown in Tab 1.

Table 1. Statistics of data set

Number of questions	6777	Avg. length before	12.22	Avg. length after	5.66
Number of comments	28911	Avg. length before	118.51	Avg. length after	55.94
Number of users	16739	Avg. votes per user	1.73	-	-

[4] http://www.debate.org/opinions/politics

4.2 Matching Questions

In constructing the BITSQG, one important step is to match questions. This involves semantically clustering (described in Sec. 2.1) and pairing up voting options (described in Sec. 2.2).

To evaluate the performance of matching questions, we use a ground truth which include equivalent questions and correct matching of stances. The ground truth is manually labeled. 300 questions are randomly selected from all questions. Two students are invited to first manually label semantically equivalent questions, then map the stances between equivalent questions. A total of 1437 question pairs are produced, 945 pairs are with the same polarities, i.e. a pro-stance is mapped to a pro-stance of another question, or a con-stance is mapped to a con-stance; and 492 pairs are with opposite polarities, i.e. a pro-stance is mapped to a con-stance.

The evaluation metric is precision, which is the ratio of correctly recognized stance pairs.

$$Precision = \frac{|R \bigcap T|}{|R|} \tag{16}$$

where R is the set of pairs recognized, T is the set of pairs in the ground truth.

We conduct the modified AP algorithm on all questions and comments. Parameter $damping = 0.5$, maximal number of iteration is set to be $N = 200$. For questions without comments, when $preference = median(S)$ we obtain 783[5] clusters, when $preference = 2 * median(S)$ we obtain 335 clusters. For questions with comments, we ... We compare AP with a parametric clustering algorithm: K-means. The comparative methods for pairing up stances include: (1) Default: which is to pair a pro-stance of a question to a pro-stance of a similar question; (2) Text: the proposed classification algorithm based on textual features f_1, f_2; (3) Vote: classification based on votes f_3, f_4; (4) Demographic: classification based on groups of votes f_5, f_6 (5) Sentiment: classification based on sentimental features f_7, f_8 (6) Combine: classification based on all 8 features.

We have the following observations from Fig. 5. (1) Textual content in comments provide valuable information. AP on questions and comments is superior (almost doubles) than AP on questions only for various settings. (2) AP can determine the best number of clusters. The performance of AP with 204 clusters is very similar to that with 121 clusters. The best performance is AP with 121 clusters on combined features is 0.6896, when there are 121 clusters, the precision is 0.6875. This shows that AP converges to the best number of clusters. (3) When the number of clusters for K-means is set to be the number of clusters obtained by AP, the performance is dramatically reduced by 40%. This again verifies that AP has the ability to self-adjusting to the optimal clustering. (4) When we set the number of clusters to be 10, the performance of Kmeans increases by roughly 40%. It suggests that the K-means needs a smaller number of clusters. However the performance is still slightly worse than the performance of AP with 204 and 121 clusters.The underlying problem is that the questions are about a wide range

[5] The number of clusters is based on the entire data set, not just the ground truth.

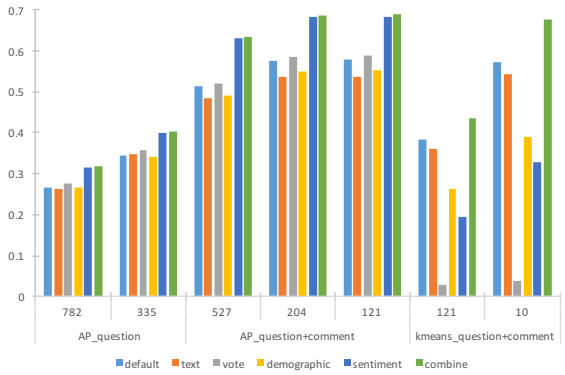

Fig. 4. Precision of various features and clustering methods for paring up stances of questions

of topics, in nature they are very dissimilar. A large number of clusters is more reasonable. (5) Combining various features always achieves the best precision. For AP methods, sentimental features alone can produce comparative performance, while for K-means methods, it is more important to combine all features than to depend on a single type of features.

4.3 Key Phrase Extraction and Weighing

We next evaluate the performance of key phrase extraction and weighing. We use Prefixspan [4] to discover question templates. We set the minimum support to $min_support = 10$. The top discovered question templates are .

- should/can ... be/to/in/of ... ?
- do you agree/think/believe/ support that ... is ... ?
- is it/this/that ... ?
- ... yes or no?

To evaluate the performance of key phrase weighing, we select a topic $o =$"gay marriage", then from the extracted key phrases, we manually score the key phrase in a scale of $1-5$. In this manner, we obtain a ranking list of key phrases. We also rank key phrases based on the weights given by Sec. 3.

The evaluation metric is NDCG, which is a standard measure for ranking systems. $NDCG = \Sigma_L NDCG@L$, where $NDCG@L$ is the normalized discounted cumulative gain for top L results.

$$NDCG@L = \sum_{i=1}^{M} \frac{\sum_{l=1}^{L}(2^{r_i(p(l,i))} - 1)/log(1+l)}{M \sum_{l=1}^{L}(2^{r_i(t(l,i))} - 1)/log(1+l)} \tag{17}$$

where $p(l,i)$ is the index of the key phrase sorted in descending order by $w(k)^o$, $t(l,i)$ is the index of the key phrase sorted in descending order by humans.

To demonstrate the performance of different coefficients w_1, w_2, we set $w_1 = 100, w_2 = w_1/x$. The NDCG with varying x is shown in Fig. 5. We can see that: (1) The key phrase weighing scheme performs good as NDCGs for all x are high, approximately 1. (2) The best performance in term of NDCG is achieved when $x = 5, NDCG = 0.9865$. (3) The sharp decrease of NDCG when $x > 5$ indicates that though comments are not as important as questions, their importance should not be ignored.

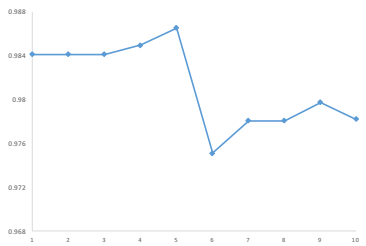

Fig. 5. NDCG of key phrase ranking on different values of x, $w_1 = 100, w_2 = w_1/x$

4.4 Evaluating the wording quality

We conduct a qualitative study of the questionnaire generated. According to the general guidelines for writing polling questions in [14], we conduct a user survey.The subjects of the survey are 3 students. Each participant is asked to read the questionnaire generated by different systems, without knowing which system generates which questionnaire. The participants must compare the results of different systems in a random order from the following point of views: (1) Coverage: does the questionnaire cover all relevant issues? (2) Diversity: are the questions diversified, e.g. each question is about one issue at a time? (3) Simpleness: does the question use simple language and is it more easily understood by respondents? (4)Concreteness: is the question clear and specific and that each respondent will be able to answer? (5)Offensiveness: are there certain words that may be viewed as biased or potentially offensive to some respondents? (6) Emotion: are there words that may provoke emotional reaction? A score of 1 to 5 needs to be assigned to each system according to the user's satisfaction of the results. For coverage, diversity, simpleness and concreteness, a rank of 5 indicates that the result of the system is the most satisfactory. For the rest two aspect, a rank of 1 indicates the best system.

As the questionnaire generation assembles the task of summarizing most informative questions, we compare our system to various summarization systems.

The comparative methods are (1) Random: questions are randomly selected. (2) Sumbasic: this algorithm greedily selects the set of questions covering the maximal weights of key phrases. (3) TextRank [8]: a graph-based ranking model for text summarization. (4) LexRank [1]: a graph-based ranking model which selects sentences with most lexical centrality. (5) LSA [3]: a text summarization framework which uses the latent semantic analysis technique to identify semantically important sentences.

Table 2. User Study of Different Questionnaire Generation Systems

	coverage	diversity	simpleness	concreteness	offensiveness	emotion
random	2.25	4.875	3.125	2.5	1.25	1.625
sumbasic	3.25	4.5	4.25	2.125	**0.625**	**0.625**
TextRank	2.875	5	3.5	3.375	1	0.875
LexRank	3.125	5	4.125	3.125	1.25	2.125
LSA	2.5	5	2.125	3.125	1.125	1.875
MIMW	**3.5**	**4.875**	**4.25**	**3.75**	0.75	0.75

As shown in Tab. 2, our system achieves best performance in coverage, diversity, simpleness and concreteness. Our system is the second best in terms of offensiveness and emotion. The user study demonstrate that the questionnaire produced by our system satisfy the general requirements for opinion polling composition.

Finally we present a case studies. The topic is "drug abuse". As shown in Tab. 3, each system selects 3 questions. The questions selected by our model are simple, clear and concrete, each covers a different aspect of the topic. On the contrary, some questions selected by other systems are confusing, i.e. asking "is it OK .." can be misleading. Some systems tend to choose long questions, i.e. Random and LSA. It could be harmful for estimating the best polling results.

5 Related Work

More and more work are devoted to mining and analyzing publicly available opinionated texts. The first line of work is to extract opinions, i.e. classifying positions and stances [10]. The second line is to summarize public opinions, i.e. generating an opinion summary [12], or visualizing the polarity of each aspect of a product [7,9]. Some works study the problems of opinion leader identification [13] , or opinion search [2]. Some recent works attempt to linking text sentiment on online social media sites to polling results [11,6]. However, polling is still essential if we prefer a more accurate and detailed opinion survey. This is a relatively unexplored area in the opinion mining literature.

The work in this paper is also related to research areas of text summarization. In particular, the polling questionnaire is similar to extractive summarization,

Table 3. Case study: questionnaire on "drug abuse"

Random	-Millionaire invests in marijuana start-up company: Should marijuana become a mainstream product and be marketed as such? -Is it ok for the government to legalize drugs? -Should employers be allowed to subject employees to mandatory drug testing?
SumBasic	- Should the U.S. legalize marijuana? - Should drugs be illegal? - Should welfare recipients be drug tested?
TextRank	- Should the recreational use of marijuana be made legal in the United States? - Should the House consider the war on drugs to be an effective policy among free people? - Should welfare recipients be drug tested?
LexRank	- Is there a good reason why marijuana is illegal and alcohol and tobacco are legal? - Should drugs be illegal? - Is it ethical to drug test welfare recipients?
LSA	- Should individual states have more or less power compared to the federal government when implementing laws dealing with legalization of marijuana gay marriage and abortion? - Has the War on Drugs really resulted in 45 million Americans being locked up for drug-related offenses at a cost of trillions? - If welfare recipients are drug tested should we also test them for alcohol
MIMW	- Should the recreational use of marijuana be made legal in the United States? - Should all drugs be legalized? - Should someone receiving welfare be drug tested?

where a concise set of units must be selected from a bunch of candidates. Many extractive summarization models are graph based [8,1,3]. They follow a general procedure, where they first construct of graph of textual units (i.e. sentences or minimal meaningful units), then they compute saliences for each textual unit on the graph. Our proposed framework is inspired by previous summarization methodologies.

6 Conclusion

In this paper we study a novel problem of questionnaire generation. We propose a framework which is based on a bi-level topic sensitive question graph. We combine semantic similarity and similarities in responses to match equivalent questions. We conduct comprehensive quantitive and qualitative experimental to verify the competency of our methods.

One possible limitation of the proposed work is that we assume that the users randomly pick a question to vote. This may be an idealized situation. Selection bias occurs when the samples are not randomly selected. If so, the response similarities are not accurate. In the future work, we will continue to work on this direction and develop methods to overcome the missing values of votes.

References

1. G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, Dec. 2004.
2. S. Gerani, M. J. Carman, and F. Crestani. Proximity-based opinion retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 403–410, New York, NY, USA, 2010. ACM.
3. Y. Gong and X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. Citeseer, 2001.
4. J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *proceedings of the 17th international conference on data engineering*, pages 215–224, 2001.
5. M. Hu and B. Liu. Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760, 2004.
6. V. Lampos, D. Preotiuc-Pietro, and T. Cohn. A user-centric model of voting intention from social media. In *Proceedings of ACL*, pages 993–1003, 2013.
7. X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 379–387, New York, NY, USA, 2012. ACM.
8. R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.
9. S. Moghaddam and M. Ester. The FLDA model for aspect-based opinion mining: addressing the cold start problem. In *Proceedings of the 22nd international conference on World Wide Web*, pages 909–918. International World Wide Web Conferences Steering Committee, May 2013.
10. A. Murakami and R. Raymond. Support or oppose?: Classifying positions in online debates from reply activities and opinion expressions. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 869–875, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
11. B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11(122-129):1–2, 2010.
12. M. Potthast and S. Becker. Opinion summarization of web comments. In *Proceedings of the 32Nd European Conference on Advances in Information Retrieval*, ECIR'2010, pages 668–669, Berlin, Heidelberg, 2010. Springer-Verlag.
13. X. Song, Y. Chi, K. Hino, and B. Tseng. Identifying opinion leaders in the blogosphere. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 971–974. ACM, 2007.
14. W. M. Trochim. *The Research Methods Knowledge Base*. Atomic Dog Publishing, Cincinnati, OH., second edition edition, 2000.