2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing

Measuring Opinion Relevance in Latent Topic Space

Wei Cheng*, Xiaochuan Ni†, Jian-Tao Sun†, Xiaoming Jin‡, Hye-Chung Kum*, Xiang Zhang§, Wei Wang*
*Department of Computer Science, University of North Carolina at Chapel Hill
Email: {weicheng,kum,weiwang}@cs.unc.edu
†Microsoft Research Asia
Email: {xini,jtsun}@microsoft.com
‡School of Software, Tsinghua University
Email: xmjin@tsinghua.edu.cn
§Department of Electrical Engineering and Computer Science, Case Western Reserve University
Email: xiang.zhang@case.edu

***Abstract*—Opinion retrieval engines aim to retrieve documents containing user opinions towards a given search query. Different from traditional IR engines which rank documents by their topic relevance to the search query, opinion retrieval engines also consider opinion relevance. The result documents should contain user opinions which should be relevant to the search query. In previous opinion retrieval algorithms, opinion relevance scores are usually calculated by using very straightforward approaches, e.g., the distance between search query and opinion-carrying words. These approaches may cause two problems: 1) opinions in the returned result documents are irrelevant to the search query; 2) opinions related to the search query are not well identified. In this paper, we propose a new approach to deal with this *topic-opinion mismatch* problem. We leverage the idea of Probabilistic Latent Semantic Analysis. Both queries and documents are represented in a latent topic space, and then opinion relevance is calculated semantically in this topic space. Experiments on the TREC blog datasets indicate that our approach is effective in measuring opinion relevance and the opinion retrieval system based on our algorithm yields significant improvements compared with most state-of-the-art methods.**

I. INTRODUCTION

With the rapid development of World Wide Web, especially the popularity of Web 2.0 applications, users are involved in many online activities and are contributing various kinds of data to the Web, for example, user opinion data. It is common practice for people to write reviews on shopping sites, forums, or blogs to express their pains and gains. Taking Amazon.com as an example, it has already organized more than one hundred million reviews written in English, and many written in other languages. Meanwhile, users like to read opinions of other people. For instance, according to Forrester research, 71% of online shoppers read reviews. Also, based on a study of 2,000 shoppers by eTailing group, 92% of them deemed customer reviews as "extremely" or "very" helpful [1]. Due to such large amounts of useful opinion data, there is a strong need to organize and access them efficiently and effectively.

Opinion retrieval is to retrieve documents containing user opinion about a search query from a text data corpus. This problem has drawn more research efforts after TREC (Text REtreival Conference) introduced a new track for this task on blog data in 2006 [2]. Since then, some algorithms have been proposed and compared [3], [4], [5], [6]. The opinion retrieval task is related to but different from traditional Information Retrieval (IR) problem. In IR, search results are ranked according to how relevant they are with the search query without considering if the documents contain query-related user opinions or not. Obviously, despite the success of search engine, it does not solve the problem of opinion retrieval. In the opinion mining area, there are already many research work in opinion identification, extraction, classification and summarization, etc. Therefore, a feasible solution which is commonly used in existing opinion retrieval work is to combine IR and opinion mining technologies. Generally, a two-stage approach is adopted. First, traditional IR technologies are used to obtain relevant documents. Secondly, opinion mining technologies are applied on those documents to identify opinions, estimate the relevance of opinion to the given query and then re-rank the documents. The unique and most challenging part of this retrieval approach is how to calculate opinion relevance of documents to the search query.

In the literature, the opinion relevance is handled by very straightforward methods. For example, in [7], the authors count the number of opinion-carrying words (like "perfect", "terrible", etc.) around search query terms and W. Zhang et al. [4] count opinion-carrying sentences (identified by sentiment classifier) around the search query. The drawbacks of this kind of solutions are obvious. Opinions around the search query may not be related to the query. Documents with irrelevant opinions would be ranked high in search results. In addition, the terms in search query cannot represent all the semantics of the search topic. Opinions which are expressed implicitly about the search topic, e.g., about some aspect of the topic, may be missed. Therefore, a very related document may be ranked very low. Basically, topic and opinion are mismatched.

In this work, we consider the document as a *bag of sentences* and propose a topic model based approach to deal with the *topic-opinion mismatch* problem. We leverage the idea of Probabilistic Latent Semantic Analysis. Both the search query and documents are represented in one latent topic space. Opinion relevance is considered at a broad topic level and calculated in the topic space. Experiments on benchmark datasets indicate that our measurement of opinion relevance is meaningful for the opinion retrieval task and our topic

978-0-7695-4578-3/11 $26.00 © 2011 IEEE
DOI

model based retrieval method can reduce the probability of *mismatch* between topic and user opinion, and the opinion retrieval system based on our algorithm is better than previous solutions.

To the best of our knowledge, we are the first to use topic model for resolving *topic-opinion mismatch* problem. The rest of this paper is organized as follows. In Section 2, we will review related works and in Section 3, we will formulate our task. The topic-based opinion retrieval framework will be presented in Section 4, followed by experiments and results analysis in Section 5. Comparisons between our approach and other state-of-the-art methods are also given in this section. Finally, we will conclude this paper and discuss future research in Section 6.

II. RELATED WORK

In the literature, opinion retrieval research communities mainly focus on three problems: opinion identification, integrating topic relevance and opinion for ranking, identifying opinion targets.

Broadly speaking, there are two groups of opinion identification algorithms: lexicon-based methods and classification-based methods [8], [4]. Lexicon-based methods use bag-of-words approach which treats a document as a collection of words without considering the relations between individual words. The sentiment of every word is determined by looking up a sentiment word dictionary. The sentiment of a sentence or a document is estimated by combining the sentiment scores of all words using some aggregation functions like average, sum, etc. Some researchers constructed the sentimental word dictionary using Web search, while others used WordNet [9] to expand a group of sentimental words [10]. M. Zhang et al. [7] used SentiWordNet [11] directly for word expansion. Some researchers used word distributions over dataset [12]. Classification-based approaches usually trained classifiers by learning from both opinion-bearing and neutral web pages using language features. These features were commonly used, like unigrams or part-of-speech data. Some researchers used domain-dependent features [8] while others used domain-independent linguistic features to determine the presence of opinion [13]. In addition, K. Seki et al. [14] introduced trigger language model to identify opinion.

Most state-of-the-art approaches used a two-stage methodology. In the first step, traditional IR techniques were used to gain topically relevant documents. In the second step, opinion identification techniques were used to get opinion strength score which is used for re-ranking those topic relevant documents. Major solutions on the combination strategy of topic relevance score and opinion score were heuristic linear combination [4] and quadratic combination [7]. In [7], M. Zhang et al. proposed that using quadratic combination to combine multiple ranking functions is superior to using a linear combination.

Few research focus on resolving the *topic-opinion mismatch* problem. Current solutions mainly use distance-based approximate methodology. They usually fix a region which might talk about the query topic, and ignore opinions outside such region. M. Zhang et al. [7] used this idea at word level and measured the strength of opinion towards the given query based on the co-occurrence of the query phrase and sentimental words within a word window. W. Zhang et al. [4] did this at the sentence level. They assumed opinion expressing sentences with distance away from the query phrase less than five sentences were relevant to the query topic. In the NLP field, [15], [16] used NLP techniques to analyze the target of sentimental adjectives. In order to capture opinion term relatedness to the query, S. Gerani et al. proposed a proximity-based opinion propagation method to calculate the opinion density at each point in a document [17].

In addition, there are also many researchers who focus on domain-specific opinion retrieval [18]. X. Liao et al. [18] used blog timestamp to help identify the relevance between opinion and the query topic. B. Liu et al. [8] made use of field-specific features of electronic products. TREC blog track also provided various research on opinion retrieval. Most of the participants of this track used blog properties in designing their solutions such as spam detection, blog structure analysis, timestamp analysis, etc. Since these solutions depend on non-textual nature of blogs, it would be difficult to generalize their methods on other types of datasets.

III. OPINION RETRIEVAL TASK

In this section, we will define some concepts related to the opinion retrieval problem and reveal its key challenges.

- Let Q stand for a search query, then the query topic of Q refers to the query itself as well as different aspects of query Q. We denote query topic of Q as Q_{topic}.
- The query topic relevant documents are the ideal output of a document retrieval system [4]. We denote Q topic relevant document as $D_{relevant\_to\_Q}$.
- The opinion expressing documents are the documents containing any types of opinions, which might not be necessarily related to the query. We denote opinion bearing document as $D_{opinion}$.
- A query relevant opinion expressing document of query Q is a document that contains at least one opinion relevant sentence about Q_{topic}. We denote Q relevant opinion expressing document as $D_{opinion\_to\_Q}$. Within $D_{opinion\_to\_Q}$, we define the opinion relevant sentence of query Q as the comment or opinion expressing sentence that is about the query topic Q_{topic}, denoted by $S_{opinion\_to\_Q}$.

For a corpus C and query Q, let $SET_{relevant\_to\_Q}$ be the set of $D_{relevant\_to\_Q}$, $SET_{opinion}$ be the set of $D_{opinion}$, and $SET_{opinion\_to\_Q}$ be the set of $D_{opinion\_to\_Q}$. Obviously, $SET_{opinion\_to\_Q}$ is a proper subset of $SET_{relevant\_to\_Q} \cap SET_{opinion}$. $SET_{relevant\_to\_Q}$ and $SET_{opinion}$ can be retrieved by using traditional IR techniques and opinion mining technologies respectively. However, simply treating $SET_{relevant\_to\_Q} \cap SET_{opinion}$ as $SET_{opinion\_to\_Q}$ will give birth to the *topic-opinion mismatch* problem since opinions shown in documents $(SET_{relevant\_to\_Q} \cap SET_{opinion}) - SET_{opinion\_to\_Q}$ are irrelevant to query topic. The crucial challenge of opinion

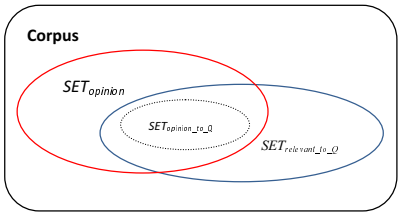

Fig. 1. Relationship among $SET_{opinion\_to\_Q}$, $SET_{relevant\_to\_Q}$ and $SET_{opinion}$

retrieval task is in fact to distinguish $SET_{opinion\_to\_Q}$ from $SET_{relevant\_to\_Q} \cap SET_{opinion}$. Figure 1 visualizes the relationship among $SET_{opinion\_to\_Q}$, $SET_{relevant\_to\_Q}$ and $SET_{opinion}$.

Example. To better illustrate our focus, we provide an example of query "iPod". Opinion retrieval task for this query aims to find documents which contain opinions about "iPod". For documents d_1 and d_2 with two paragraphs respectively, the first paragraph of d_1 is talking about "iPod" but does not contain any opinions. While the second paragraph of d_1 is talking about "iphone" and also has opinions or sentimental expressions about "iphone". Then d_1 will not be the ideal output of an opinion retrieval system for query "iPod" in that it does not contain opinions about the query topic. For d_2, its first paragraph narrates the objective aspects of "iPod", and it shows opinions about "iPod" in the second paragraph. This describe-first-then-comment mode is a common expression form in user reviews. If the opinion sentences in the second paragraph of d_2 do not contain query term "iPod", then these opinion expressing sentences should not be treated as query irrelevant and d_2 is an ideal output of the opinion retrieval system.

IV. Topic Model Based Opinion Retrieval

Most state-of-the-art solutions for opinion retrieval adopt a two-stage methodology. In the first step, traditional IR techniques are used to obtain topically relevant documents. In the second step, opinion identification techniques are applied to estimate opinion score which is used for re-ranking the topically relevant documents. In this work, we also follow this two-stage framework and mainly focus on how to better estimate opinion relevance scores.

To determine whether opinion expressing sentences are relevant to Q_{topic}, most solutions in literature are simply based on the distance between exactly or partially matched query terms (or phrases) and opinion expressing sentences. This simple treatment, however, may cause the *topic-opinion mismatch* problem [19]. Essentially, previous solutions do not treat query as a topic concept which may have multiple aspects. For an example of query "March of the Penguin", if the opinion retrieval engine understands that the query is talking about a movie and its related topical aspects may include actress, script, director and story, etc., the risk of mismatching opinion with topic will be reduced. In this work, we try to leverage the topic model to help boost the performance of opinion retrieval. The framework can be briefly summarized as follows:

1) Use traditional IR method to search for a group of documents which are topically relevant to the search query, regardless of whether they contain user opinions or not;

2) Apply a topic modeling algorithm, Probabilistic Latent Semantic Analysis (PLSA), to infer the topic space from the top-ranked documents obtained in the previous step;

3) Project the search query and opinion carrying sentences included in the search result documents into the topic space. Estimate opinion relevance scores between the query and each result document;

4) Re-rank initial search documents based on topic relevance scores and opinion relevance scores estimated in the previous steps.

A. Retrieve Topically Related Documents

In order to infer a topic space within which we measure opinion relevance between document and search query, we first use traditional IR methods to retrieve a group of documents. Here, we only consider if a document is relevant to the search query, regardless of whether the document contains user opinion or not. The reason is that we hope to obtain a group of documents which can be used to infer different aspects related to the search query (query topic). Opinion relevance score will be estimated in subsequent steps. In this work, the BM-25 [20] IR method is used to calculate the relevance between query and documents:

$$Score I_{IR} = \sum_{t \in Q,D} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1(1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf} \quad (1)$$

where tf is the term's frequency in document, qtf is the term's frequency in query, N is the total number of documents in the collection, df is the number of documents that contain the term, dl is the document length (in bytes), $avdl$ is the average document length, k_1 (between 1.0-2.0), b (usually 0.75), and k_3 (between 0-1000) are constants.

The top M (M is a parameter) relevant documents, regarded as the working set, will be used for estimating topic space. We use BM-25 due to its easy implementation and popularity. Other IR approaches like language model can also be used to obtain the working set.

B. Infer Topic Space

After we obtain the working set, we will use Probabilistic Latent Semantic Analysis (PLSA) for inferring a topic space. PLSA [21] models each document as a mixture of topics, each being a unigram model. It generates documents with the following process:

1) Select a document d with probability $p(d)$

2) Pick a latent topic z with probability $p(z|d)$

3) Generate a word w with probability $p(w|z)$

If the document is treated as a bag of words $w_1, \ldots, w_n$ (n is the number of words in the document set), then a document d_i in the working set can be regarded as a sample of the following

mixture model:

$$p(w_j|d_i) = \sum_{k=1}^{K} p(w_j|z_k)p(z_k|d_i) \tag{2}$$

where z_k is the k-th topic, and $p(z_k|d_i)$ is a document-specific mixing weight for the k-th topic, and if there are K topics in total, we have:

$$\sum_{k=1}^{K} p(z_k|d_i) = 1 \tag{3}$$

The log-likelihood of the working set W is

$$\log p(W|\Lambda) = \sum_{d_i \in W} \sum_{w_j \in V} n(d_i, w_j) \log p(w_j|d_i) \tag{4}$$

where Λ is the set of the all model parameters, and V is the set of all words (i.e., vocabulary), $n(d_i, w_j)$ is the count of the word in the document d_i. This model can be estimated with any estimator. One of the estimators widely used for estimating parameters for PLSA is Expectation-Maximization (EM) algorithm [22] with the following updating formulas:

E-step:

$$p(z_k|w_j, d_i) = \frac{p(w_j|z_k)p(z_k|d_i)}{\sum_{k=1}^{K} p(w_j|z_k)p(z_k|d_i)} \tag{5}$$

M-step:

$$p(w_j|z_k) = \frac{\sum_{i=1}^{M} n(d_i, w_j)p(z_k|d_i, w_j)}{\sum_{i=1}^{M} \sum_{j=1}^{|V|} n(d_i, w_j)p(z_k|d_i, w_j)} \tag{6}$$

$$p(z_k|d_i) = \frac{\sum_{j=1}^{|V|} n(d_i, w_j)p(z_k|d_i, w_j)}{\sum_{j=1}^{|V|} \sum_{k=1}^{K} n(d_i, w_j)p(z_k|d_i, w_j)} \tag{7}$$

We use PLSA to infer topic space because it is widely used to model text documents in previous research. As we know, PLSA has two drawbacks: 1) it is not efficient when there are a large number of documents to model; 2) it is not suitable for inferring topics for unseen documents. In this work, the working set usually contains at most thousands of documents. According to our experiments, PLSA is efficient given document set of this scale. Meanwhile, the topic space inferred by PLSA is used to compute opinion relevance between the query and the document, thus we do not need to infer topics for unseen documents in this work.

C. Project Search Query and Opinion Carrying Sentences into Topic Space

From Section III, the task of opinion retrieval is to find documents which contain sentences with user opinion towards query topic. If we treat each sentence as an independent document, then in traditional language modeling approach [23], the relevance between sentence s and query q can be denoted by the conditional probability $p(s|q)$. According to Bayes' formula and dropping a document-independent constant, we have

$$p(s|q) \propto p(q|s)p(s) \tag{8}$$

In this equation, $p(s)$ is our prior belief that s is relevant to any query and $p(q|s)$ is the query likelihood given the sentence, which captures how well the sentence s "fits" the particular query q. Most existing works [24] assume that $p(s)$ is uniform thus does not affect the overall measurement. In order to get $p(q|s)$, in the simplest case, exactly or partially match-based methods are often used, e.g. M. Zhang et al. [7] use exact match of query phrase within a window region. These strategies are all based on the literal match, regardless of the semantic and topic relevance. Rather than to literally match query phrase, we consider every opinion expressing sentence to be relevant to query q with a probability, and denote this probability as $p(Q_{topic}|s_{op})$, where s_{op} denotes any opinion expressing sentence in document D. Here we use Q_{topic} rather than q itself in order to emphasize that the relevance between opinion and query that we are trying to measure is at topic level rather than literal level. $p(Q_{topic}|s_{op})$ captures how well the opinion expressing sentence "fits" the particular query topic Q_{topic}.

In order to measure $p(Q_{topic}|s_{op})$, we can first project query q and opinion expressing sentences into a K dimensional topic space inferred above so that we can measure relevance between the query and opinion at the topic level. With the parameters estimated in PLSA model, we can utilize the model parameters to measure the probability of generating q and s_{op} [25]. For any opinion expressing sentence s_{op}, its corresponding vector in K dimensional topic space will be

$$\overrightarrow{p(s_{op}|z)} =< p(s_{op}|z_1), \ldots, p(s_{op}|z_K) > \tag{9}$$

where $p(s_{op}|z_k)$ is the probability for topic z_k to generate s_{op} [25].

$$p(s_{op}|z_k) = \frac{1}{K'} \cdot \sum_{j=1}^{|V|} n(s_{op}, w_j)p(w_j|z_k) \tag{10}$$

Here, K' is the normalization parameter which is equal to $\sum_{k=1}^{K} \sum_{j=1}^{|V|} n(s_{op}, w_j)p(w_j|z_k)$. Also, we project query into topic space with similar method, and the corresponding vector of q in topic space is denoted by $\overrightarrow{p(q|z)}$.

$$\overrightarrow{p(q|z)} =< p(q|z_1), \ldots, p(q|z_K) > \tag{11}$$

where $p(q|z_k)$ is the probability for topic z_k to generate query q.

$$p(q|z_k) = \frac{1}{K''} \cdot \sum_{j=1}^{|V|} n(q, w_j)p(w_j|z_k) \tag{12}$$

Similarly, K'' is the normalization parameter which is equal to $\sum_{k=1}^{K} \sum_{j=1}^{|V|} n(q, w_j)p(w_j|z_k)$.

D. Opinion Relevance Measurement at the Topic Level

The projecting method enables us to compare the similarity between sentences and queries at topic level. Since we have gained the projection vector with formula 9 and 11, we can measure the similarity between $\overrightarrow{p(q|z)}$ and $\overrightarrow{p(s_{op}|z)}$ with various metrics, such as Lp distance and Kullback-Leibler Divergence (KLD), we choose cosine similarity to measure

this relevance since cosine similarity is widely used in the text and information retrieval area. Thus, the similarity between opinion expressing sentence s_{op} and query q will be

$$\begin{aligned} & p(q|s_{op}) \\ &= p(Q_{topic}|s_{op}) \\ &= cos(\overrightarrow{p(q|z)}, \overrightarrow{p(s_{op}|z)}) \\ &= \frac{\overrightarrow{p(q|z)} \cdot \overrightarrow{p(s_{op}|z)}}{\|\overrightarrow{p(q|z)}\|_2 \times \|\overrightarrow{p(s_{op}|z)}\|_2} \end{aligned} \tag{13}$$

Note that $p(Q_{topic}|s_{op})$ is not a strict probability distribution, it's used to denote the score of the similarity between s_{op} and Q_{topic}. As we will see later, it will be used to calculate the overall score for ranking.

E. Re-ranking

After getting relevance between opinion expressing sentences and query topic, we can judge if an opinion expressing sentence is relevant to the query topic by introducing a threshold μ. In other words, those opinion expressing sentences with relevance to query topic above this threshold are determined to belong to $S_{opinion\_to\_Q}$. Then we can get the opinion score for each document, denoted by $ScoreI_{OP}$. However, how to use identified opinions to calculate documents' opinion score is still worthy of further study. Current solutions are almost empirical. Some researchers use the sum of identified opinions (sentences or sentimental words) [4], some use the density-based method like M. Zhang [7]. Since this is not our focus, we will follow the method used by W. Zhang et al. [4] which uses the sum of score for each $S_{opinion\_to\_Q}$. For a given document d, if we denote S_o to be the set of opinion expressing sentences for d, and each of these sentences has an opinion-bearing strength (estimated from classifier or sentimental lexicon), we use function $f(s)$ to measure the opinion-bearing strength of sentence s, then we can get $ScoreI_{OP}$ of document d with the formula below.

$$\begin{aligned} & ScoreI_{OP} = \textstyle\sum_{s_m} p(q|s_m) \cdot f(s_m) \quad s.t. \quad (1) \quad and \quad (2), \quad with: \\ & (1) \quad p(q|s_m) > \mu \\ & (2) \quad s_m \in S_o \end{aligned} \tag{14}$$

Certainly, $ScoreI_{OP}$ will be normalized over all initially retrieved documents so that $\widehat{ScoreI_{OP}} \in [0, 1]$, where $\widehat{ScoreI_{OP}}$ is the normalized value of $ScoreI_{OP}$. The overall score of a document is based on some forms of combination, e.g., linear combination and quadratic combination of two scores: $ScoreI_{IR}$ and $ScoreI_{OP}$, where $ScoreI_{IR}$ is the relevance score in the initial retrieval. We denote this overall score as

$$Score = Combination(ScoreI_{IR}, \widehat{ScoreI_{OP}}) \tag{15}$$

There are mainly two combination methods in combining $ScoreI_{IR}$ and $\widehat{ScoreI_{OP}}$: 1) linear combination $\lambda \cdot ScoreI_{IR} + (1-\lambda) \cdot \widehat{ScoreI_{OP}}$ [12], [4]; 2) quadratic combination $ScoreI_{IR} \cdot \widehat{ScoreI_{OP}}$ [7]. With this overall score, we can re-rank those initially retrieved topically related documents.

V. Evaluation

We evaluated our framework on a corpus provided by TREC (Text REtreival Conference) which is widely used for evaluation in opinion retrieval task. The goal is to see the effectiveness of our framework:

1. on the identification of the query topic related opinion;
2. on the ranking of query relevant opinion expressing documents.

A. Data Set and Evaluation Metrics

In order to test our framework, we use TREC Blog06 [2] corpus as our dataset considering that it is the most authoritative opinion retrieval dataset available and most recent research on opinion retrieval use this dataset for evaluation. This corpus is crawled over a period of 11 weeks (December 2005 - February 2006). The total size of this collection is 148GB with three components: feeds, permalinks and homepages. We also focus on retrieving permalinks from this dataset in consideration of two facts: (1)most current research based on that dataset for opinion retrieval choose to use permalinks only; (2)human evaluation results which are crucial for evaluating our framework is only available for permalink documents. We use 150 queries from Blog06, Blog07 [26], and Blog08 [27] for evaluation. 50 topics (Topic 851-900) are provided in Blog Track 06 and 50 topics (Topic 901-950) are also provided in Blog Track 07. Blog Track 08 provided 150 topics (Topic 851-950, and Topic 1001-1050). In order to better evaluate our framework, we will tune algorithm parameters with Blog Track 06 topics while using Blog Track 07 and 08 new topics as the testing data. In order to better compare our algorithm with other state-of-the-art solutions, we choose to use title-only query as our engine input so that the comparison is fair. The evaluation metrics used in our experiment are MAP (mean average precision), P@10 (precision at top 10 results) and R-prec (R-Precision).

B. Experiment Environment Setting

1) Sentiment Identification Policy: In order to determine if a sentence is expressing some opinion, there are two main kinds of approaches for opinion identification: lexicon-based approach and classification-based approach [8], [4], [13]. In addition, K. Seki et al. [14] introduce a subjective trigger pattern to identify opinions. Since sentiment identification is not our focus, for simplicity, we will follow M. Zhang et al. [7] and use sentiwordNet [11] to choose opinion-bearing words with the constraint that the sentiment (positive or negative) strength of a word is more than 0.6. With this constraint, we construct our opinion-bearing vocabulary V_{op}, which contains 2,371 positive words and 5,199 negative words. And we use this rule to identify opinion expressing sentences, for sentence s:

$$f(s) = \begin{cases} 1 & if \quad \exists t \in V_{op} \quad \wedge \quad s \quad contains \quad t \\ 0 & else \end{cases} \tag{16}$$

2) Combination Policy: Since our focus is on the calculation of $ScoreI_{OP}$, in the following experiment, we choose linear combination. We will study the influence of λ on the final ranking results, which enables us to better understand the role of $\widehat{ScoreI_{OP}}$ in opinion retrieval.

3) Size of the Working Set: In order to learn topic model, we choose to use pseudo feedback documents pooled as our working set for the training model. Since the more documents are used, the more time-consuming it will be to train model, there should be a trade-off. In our experiments, we empirically set the size of working set M=1000 in consideration that our evaluation is based on TREC Blog Track which evaluates top-1000 retrieval results.

C. The Baseline Methods and Implementations

Below is the setting of the three baseline implementations, and we denote S_o as the opinion expressing sentence set of document d and $f(s_m)$ denotes the sentiment strength of opinion expressing sentence s_m.

(1)**bag-of-words-Method**: This method simply treats document as a bag of words. It does not consider if the opinion is relevant to the query topic. In fact, this method simply treats $SET_{relevant\_to\_Q} \cap SET_{opinion}$ as $SET_{opinion\_to\_Q}$. $ScoreI_{OP}$ of d is calculated with the formula below:

$$ScoreI_{OP} = \sum_{s_m \in S_o} f(s_m) \tag{17}$$

(2)**single-sentence-Method**: This method only considers opinion expressing sentences that contain query phrase, and $ScoreI_{OP}$ of d is calculated with the formula:

$$\begin{aligned} &ScoreI_{OP} = \textstyle\sum_{s_m} f(s_m) \quad s.t. \quad (1) \quad and \quad (2), \quad with: \\ &(1) \quad s_m \in S_o \\ &(2) \quad s_m \quad contains \quad q \end{aligned} \tag{18}$$

(3)**window-Method**: This method considers both opinion expressing sentences containing query phrase and those opinion expressing sentences near the query. $ScoreI_{OP}$ of d will be calculated with the formula:

$$\begin{aligned} &ScoreI_{OP} = \textstyle\sum_{s_m} f(s_m) \quad s.t. \quad (1) \quad and \quad (2), \quad with: \\ &(1) \quad s_m \in S_o \\ &(2) \quad \exists s_i \in S_{neighbors\_of\_s_m} \wedge s_i \quad contains \quad q \end{aligned} \tag{19}$$

where $S_{neighbors\_of\_s_m}$ is the set of sentences which are neighbors of sentence s_m within a given distance. This method, in fact, treats sentences within the window around the query to be relevant to the query. The window size used in our experiment is set to 5 sentences, which was also chosen by W. Zhang et al. [4].

D. Results and Analysis

1) Topic Number Setting: Selecting the proper number of topics (K) is also important in topic modeling. Usually, a range of 50 to 300 topics is typically used in the topic modeling literature [28]. In order to tune the topic number, we empirically set relevance threshold μ to be 0.6 then tune the topic number. Figure 2 shows the evolution of MAP

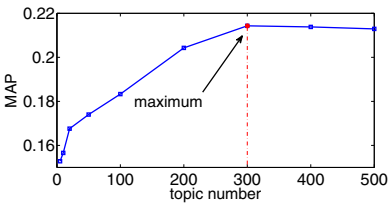

Fig. 2. MAP evolution over topic number

1 I got a second job at Ritz Camera Shop. 2 I'll be getting many more hours, and making seven dollars an hour plus commission on anything I sell, so probably about 10 dollars an hour. 3 *It'll be quite nice working with stuff that I'm into.* 4 *My brother has pink eye and he's, no doubt, been probing that thing all day whilst sitting here, typing on these very keys.* 5 Oh, Casey and I went to see March of the Penguins last night. 6 Penguins are remarkable. 7 I kept making a lot of little noises all throughout the film because it was pretty much killing me. 8 A penguin documentary? 9 *I'm so happy for you that you got that job!* 10 *It'll be so good for you since you are into photography.* 11 I wish I could get a job at a gallery or something. 12 *That would be helpful down the line.* 13 Congratulations on your new job! 14 I wanted to see march of the penguins... 15 I heard it was super good, the documentary seems wonderful. 16 The film was dubbed by Morgan Freeman. 17 I was just dropping by and saying hi. 18 I haven't seen you in forever.

Fig. 3. Test document d

corresponding to different topic numbers for the Blog Track 06 query set. From Figure 2, we find when topic number increases from 5 to 20, the MAP increases rapidly. After that, MAP increases slowly with the increase of topic number and reaches its maximum at 300 topics. Thus we can obtain a good MAP with a small topic number (e.g., 300 is used in our experiment).

2) Effectiveness of Our Framework: We provide a running example to illustrate the effectiveness of our framework. We use one document in the corpus to show why our method works. The content of the example document d is shown in Figure 3. There are 18 sentences containing both relevant and irrelevant opinions on query "March of the Penguins" which is a name of a documentary film dubbed by Morgan Freeman. These sentences are marked by number. The corresponding topical relevance values of these sentences calculated by our framework are shown in Figure 4. In the figure, both green (corresponding to sentences with italic font) and red (corresponding to sentences underlined) points are identified as opinion-bearing, while only red points are relevant opinions. We can see our framework well distinguishes those relevant opinions from irrelevant opinions and this is crucial for re-ranking initially retrieved documents.

For better evaluating our framework, three opinion retrieval baseline methods are compared. Their performances on 150 queries of Blog Track 06, 07 and 08 are shown in Figure 5. From Figure 5, we can see our framework yields better MAP and P@10 than three baseline methods. Note the smaller the value of λ, the more influence $ScoreI_{OP}$ will have on

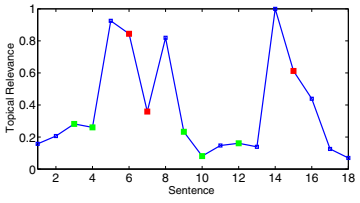

Fig. 4. Topical relevance of document d

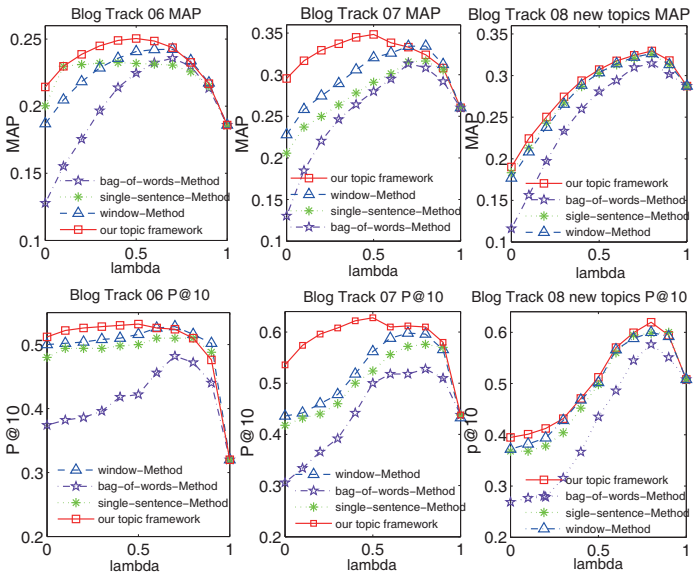

Fig. 5. MAP & P@10 comparison for Blog Track 06, 07 and 08 topics with different combination parameters

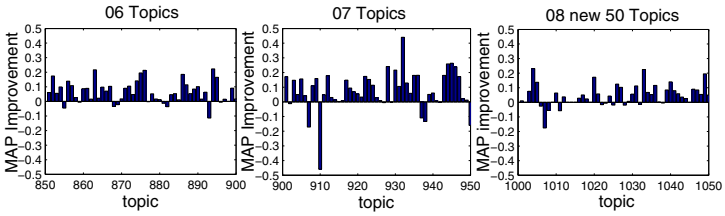

Fig. 6. MAP improvement after re-ranking for individual 150 topics of Blog Track 06, 07 and 08

the combined overall score. When λ is 1, there will be no opinion identified. Since each method has the same initially retrieved documents, the four methods have the same MAP and P@10 when λ goes to 1. We can also observe that our framework has more advantages over baseline methods when λ is small. Note that our framework and baseline methods use the same sentiment identification methods (using SentiWordNet [11], [7]). Figure 5 well indicates our framework outperforms baseline methods. In addition, the figure also shows that bag-of-words method, which identifies opinion expressions without considering opinion relevance, has poor performance. Generally speaking, window-based method [4] outperforms single-sentence method, and both methods outperform bag-of-words method. This indicates the importance of calculating the opinion relevance of the identified opinions in the opinion retrieval task. In other words, the opinion relevance provides meaningful guidelines to better measure the opinion score. Figure 6 presents MAP improvement after re-ranking using our framework. Parameter lambda (λ) was fixed to the optimum (0.5) identified above. The results for most 150 queries in Blog Track06, 07 and 08 got notable increases.

3) Comparison with State-of-the-art Methods: In order to further demonstrate the effectiveness of our framework, we also compare our framework with previous work. As we all know, most current solutions of opinion retrieval task use a two-stage method. In the first stage, traditional IR techniques are used to initially retrieve topically relevant documents. In the second stage, they re-rank those initially retrieved documents by opinion identification techniques. Thus, different baselines (initially retrieved documents) always yield different final opinion retrieval performance. Better IR engine and pre-processing (like spam filtering) often gives better input for identifying opinion in the second step. Consequently, it is wiledly agreed [7], [14] that the most important factor is to what extent the system can enhance opinion metrics (MAP, P@10, R-prec) from relevant baseline. The improvement will be the most important evaluator for judging the effectiveness of the opinion retrieval method. Thus, our comparison will focus on the improvement. In the opinion retrieval research community, M. Zhang et al. and K.Seki et al. [7], [14] also chose to focus on the improvement for evaluation. Table 1 compares results from different approaches. Since there were no improvement results reported for Blog06, we only present the evaluation results (MAP, P@10, R-prec) in Blog06. Some work has not reported their improvements over P@10 and R-prec. we only list the improvements over MAP for their work.

Table I shows that our framework yields more improvements over relevant baseline than most state-of-the-art methods. Note

TABLE I
COMPARISON OF OPINION RETRIEVAL PERFORMANCE

Data set	Method	MAP	P@10	R-prec
Blog 06	Best title-only-run at Blog06 [2]	0.1885	0.512	0.2771
	Our Relevant Baseline	0.186	0.32	0.2623
	Our framework	0.2504	0.532	0.3169
	M. Zhang et al. improvement[7]	28.38%	44.86%	16.00%
	K. Seki et al. improvement[14]	22.00%	–	–
	Our framework's improvement	**34.62%**	**66.25%**	**20.82%**
Blog 07	Most Improvement at Blog07 [26]	15.90%	21.60%	8.60%
	M. Zhang et al. improvement[7]	28.10%	40.30%	19.90%
	Our Relevant Baseline	0.2603	0.438	0.3131
	Our Framework	0.3484	0.628	0.3869
	Our framework's improvement	**33.84%**	**43.38%**	**23.57%**
Blog 08 new topics	Most Improvement at Blog08[27]	**31.60%**	–	–
	Secondary best Improvement at Blog08	14.75%	–	–
	Our Relevant Baseline	0.2818	0.498	0.3451
	Our Framework	0.3296	0.6196	0.3789
	Our framework's improvement	16.96%	24.42%	9.79%

that all these methodologies ignore the importance of opinion relevance, our topic framework is able to further enhance performance of these approaches. Since TREC Blog Track 08 queries contain both Blog Track 06 and 07 queries, we compare 50 newly added topics in Blog Track 08 separately.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel framework to measure opinion relevance in the latent topic space. In our framework, we train a topic model with the top-M pseudo feedback documents, project opinions expressed in documents into a topic space and measure opinion relevance in the topic space. Since our framework does not rely on any specific nature of the method for retrieving primitive topic relevant documents and opinion identification algorithms, it is essentially a general framework for opinion retrieval and can be used by most current solutions to further enhance their performance. Also our framework is domain-independent, thus fits for different opinion retrieval environments. The effectiveness and advantage of our framework were justified by experimental results on TREC blog datasets. According to the experiments, our framework yields much better results than both baseline experiments and state-of-the-art solutions.

In order to further enhance opinion retrieval solution, we will focus our work on two directions: 1) exploring one-step solutions which go beyond merely document re-ranking approaches, and 2) studying learning to rank methods for opinion retrieval.

REFERENCES

[1] *http://www.ratepoint.com/resources/industrystats.html*.

[2] I. Ounis, M. de Rijke, C. Macdonald, G. A. Mishne, and I. Soboroff, "Overview of the trec-2006 blog track," in *TREC'06 Working Notes*, 2006, pp. 15–27.

[3] J. Skomorowski and O. Vechtomova, "Ad hoc retrieval of documents with topical opinion," *Advances in Information Retrieval*, pp. 405–417, 2007.

[4] W. Zhang, C. Yu, and W. Meng, "Opinion retrieval form blogs," in *CIKM'07*, 2007, pp. 831–840.

[5] D. J. Osman and J. L. Yearwood, "Opinion search in web logs," in *ADC'07*, 2007, pp. 133–139.

[6] B. He, C. Macdonald, and I. Ounis, "Ranking opinionated blog posts using opinionfinder," in *SIGIR'08*, 2008, pp. 727–728.

[7] M. Zhang and X. Ye, "A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval," in *SIGIR'08*, 2008, pp. 411–418.

[8] B. Liu, M. Hu, and J. Cheng, "Opinion observer: Analyzing and comparing opinions on the web," in *WWW'05*, 2005, pp. 342–351.

[9] C.Fellbaum, "Wordnet: An electronic lexical database (language, speech and communication)," *The MIT Press*, May 1998.

[10] S. Kim and E. Hovy, "Determining the sentiment of opinions," in *COLING'04*, 2004, pp. 1367–1373.

[11] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *LREC'06*, 2006, pp. 417–422.

[12] G. Mishne, "Multiple ranking strategies for opinion retrieval in blogs," in *TREC 2006*, 2006.

[13] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," *CoRR, cs. CL/0205070*, 2002.

[14] K. Seki and K. Uehara, "Adaptive subjective triggers for opinionated document retrieval," in *WSDM'09*, 2009.

[15] M. Mulder, A. Nijholt, M. D. Uyl, and P. Terpstra, "A lexical grammatical implementation of affect," in *Proceedings of Text Speech and Dialogue-2004, the7th.International.Conference*, 2004, pp. 171–177.

[16] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques," in *ICDM'03*, 2003.

[17] S. Gerani, M. J. Carman, and F. Crestani, "Proximity-based opinion retrieval," in *SIGIR'2010*. ACM, 2010, pp. 403–410.

[18] X. Liao, D. Cao, S. Tan, Y. Liu, G. Ding, and X. Cheng, "Combining language model with sentiment analysis for opinion retrieval of blogpost," in *TREC'06*, 2006.

[19] H. Fang and C. Zhai, "Semantic term matching in axiomatic approaches to information retrieval," in *SIGIR'06*, 2006, pp. 115–122.

[20] A. Singhal, "Modern information retrieval: a brief overview," in *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2001, pp. 35–42.

[21] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR'99*, 1999, pp. 50–57.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of Royal Statist*, pp. 39:1–38, 1977.

[23] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *SIGIR'01*, 2001, pp. 334–342.

[24] A. Berger and J. Lafferty, "Information retrieval as statistical translation," in *SIGIR'99*, 1999, pp. 222–229.

[25] Y. Lu and C. X. Zhai, "Opinion integration through semi-supervised topic modeling," in *WWW'03*, 2008.

[26] C. Macdonald, I. Ounis, and I. Soboroff, "Overview of the trec-2007 blog track," in *TREC'07*, 2007.

[27] I. Ouni, C. Macdonald, and I. Soboroff, "Overview of the trec-2008 blog track," in *TREC'08*, 2008.

[28] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, pp. 993–1022, 2003.