Proximity-Based Opinion Retrieval

Shima Gerani
University of Lugano
Faculty of Informatics
Lugano, Switzerland
shima.gerani@usi.ch

Mark J. Carman
University of Lugano
Faculty of Informatics
Lugano, Switzerland
mark.carman@usi.ch

Fabio Crestani
University of Lugano
Faculty of Informatics
Lugano, Switzerland
fabio.crestani@usi.ch

ABSTRACT

Blog post opinion retrieval aims at finding blog posts that are relevant and opinionated about a user's query. In this paper we propose a simple probabilistic model for assigning relevant opinion scores to documents. The key problem is how to capture opinion expressions in the document, that are related to the query topic. Current solutions enrich general opinion lexicons by finding query-specific opinion lexicons using pseudo-relevance feedback on external corpora or the collection itself. In this paper we use a general opinion lexicon and propose using proximity information in order to capture opinion term relatedness to the query. We propose a proximity-based opinion propagation method to calculate the opinion density at each point in a document. The opinion density at the position of a query term in the document can then be considered as the probability of opinion about the query term at that position. The effect of different kernels for capturing the proximity is also discussed. Experimental results on the BLOG06 dataset show that the proposed method provides significant improvement over standard TREC baselines and achieves a 2.5% increase in MAP over the best performing run in the TREC 2008 blog track.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation, Performance

Keywords

Opinion, Sentiment, Blog, Retrieval, Proximity

1. INTRODUCTION

Blog post opinion retrieval is the problem of finding blog posts that express opinion about a given query topic. This problem was introduced in the Text REtrieval Conference (TREC) 2006 blog track and continued to 2008 [15, 12, 16]. The proposed approaches mostly follow a three step framework. In the first step, traditional IR is used to find documents that are relevant to the query. In the second step, opinion scores are generated for the relevant documents. Finally, a ranking method is used to rank documents according to their relevance and opinionatedness about the query.

Blog post opinion retrieval faces two main challenges. The first challenge is to find the best way to combine relevance and opinion scores to produce a single ranking. In previous work, researchers mostly used linear combinations of relevance and opinion scores [16]. We use a probabilistic approach and propose a simple model for combining probabilities of relevance and opinionatedness about a query.

The second challenge is assigning query-related opinion scores to documents. The problem is how to identify opinion expressions in the document that are directed at the concepts in the query. Simple averaging over the opinion weights of terms or sentences in a document to generate an opinion score is not an optimal approach. The reason is that documents can be relevant to many different topics at the same time but the opinion being expressed in them may be directed towards topics other than the query. So we need an opinion finding method that takes the query into account and ignores opinionated content that is not related to the query. In this paper we propose using proximity-based density functions to model the notion of query-relatedness for opinionated content. Our aim is to see how much improvement can be achieved using proximity information alone without the need for query-specific opinion-lexicon. Our contributions are:

- Presenting a novel probabilistic opinion retrieval model that is based on proximity between opinion lexicons and query terms.
- Investigating different ways of estimating the relevance probability of documents from their relevance scores.
- Investigating the impact of different types of proximity functions in our model.

We evaluate our model on the BLOG06 collection using five standard TREC 2008 baselines. We report significant improvements over these strong TREC baselines and over non proximity-based opinion retrieval scores in the experiments. The results show the effectiveness of utilizing the simple proximity information in enhancing opinion retrieval, compared to systems which utilize components such as query

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

expansion and query-specific opinion-lexicon refinement via pseudo-relevance feedback.

2. RELATED WORK

Research on opinion mining and sentiment analysis started mostly on review-type data with the intention to classify documents as expressing either a positive or negative opinion. The proposed approaches can be categorized into two main groups: lexicon-based [21, 20, 24] and classification-based [17, 1, 13, 5]. Both of these approaches rely on word occurrences. The first approach (lexicon based), uses a manually or automatically built list of subjective words, such as 'good' and 'like', and assumes that the presence of these words in a document is the evidence of document opinionatedness. A term's opinion score can be used in different ways to assign an opinion score to the whole document. The second approach (classification-based) utilizes word occurrence and sometimes linguistic features and builds a classifier based on positive (opinionated) and negative (non-opinionated) documents using Machine Learning techniques. Nevertheless, most of the early research in this area neglected the problem of retrieving documents that are related to the topic of the user's interest. It also did not target the problem of ranking opinionated documents according to the degree with which they are opinionated (either in positive or negative way). Relevance of an opinion to a topic was considered for the first time in Yi et. al [24] and then in Hurst and Nigam's work [7] but they did not consider the ranking of documents. Instead they only classified documents as to whether they expressed an opinion about the topic. Opinion ranking was considered for the first time in Eguchi and Lavrenko's work [2].

In TREC 2006, the *Opinion Retrieval* task appeared in the Blog track. It considered the opinion mining problem from an Information Retrieval perspective and introduced the problem of retrieving and ranking blog posts that were relevant and opinionated about a given topic (query) [15]. There has been lots of research on blog opinion retrieval in TREC [15, 12, 16] and other conferences [26, 25, 19] in which people follow the opinion retrieval definition used in the TREC blog track. Following the categorization mentioned earlier, the proposed methods belong to two classes of lexicon-based [23, 25, 6, 9, 19] and classification-based [26, 8] approaches and usually follow the three-step framework mentioned earlier.

In this paper we follow the TREC opinion retrieval problem definition. We follow the lexicon-based approach in opinion finding and focus on the problem of finding topic related opinion expressions. In the rest of this section we explain in greater detail relevant previous work in handling the opinion retrieval challenge.

2.1 Capturing Topic Related Opinion Expression

Since the aim of opinion retrieval is to find documents that express an opinion about the query, unrelated opinion should not be considered in the scoring of a document. Therefore, the main challenge in opinion finding is to score documents by opinion expressions that refer to the query. In previous work, researchers followed two orthogonal approaches. In the first approach they built a query-specific opinion lexicon by starting from a general opinion lexicon and refining the opinion weights of terms in the lexicon via feedback style learning on the top retrieved documents in response to the query [9, 14].

The second approach uses the proximity of subjective terms or sentences to the topic terms as a measure of relatedness [1, 26, 25, 19, 22]. Dave et al. [1] tried to capture proximity using higher order n-grams as units to represent text. However, n-grams cannot capture the dependency of non-adjacent terms. Although such dependencies can be captured by increasing the length of the n-gram, this can be impractical due to a lack of sufficient training data. Zhang et al. [25] calculate the proximity of opinion terms to query terms by computing the probability of query term and opinion term co-occurrence within a window. Vechtomova [22] considered the distance in the number of non-stopwords between a query term and subjective lexical units occurring within the window of n words around the query term. Although they considered proximity information in their models, Zhang et al. [25] did not find any advantage of using the proximity information while Vechtomova [22] did show some improvement in terms of opinion MAP. In this paper we introduce the proximity information in a more principled way and show that it can improve the performance over a non proximity-based opinion retrieval baseline.

Proximity information is also considered in [26, 19], with the difference being that they first find opinionated sentences and then consider proximity of opinionated sentences to the query term. Zhang et al. [26] use a SVM classifier to classify document's sentences as either opinionated or non-opinionated. They then apply a NEAR operator to classify an opinionated sentence as either relevant or not relevant to the query. Santos et al. [19] use a divergence from randomness proximity model to integrate the proximity of query terms to the opinionated sentences identified by a general opinion finding system. They further combine the proximity scored opinion sentences by the relevance score of the document using a linear combination. Our work is similar to this method in the sense that we also use a general opinion lexicon without refining it with query specific opinion terms, but our method differs in that we do not work on the sentence level but use the opinion weights and proximity of terms to the query directly. We also consider a proximity-based opinion density functions to capture the proximity information that has not been used in previous studies in opinion retrieval. The way we incorporate the relevance score in our model is also different from the previous studies in that we investigate different ways of estimating the relevance probability from the document's relevance score.

2.2 Combining Relevance and Opinion Scores

In order to produce a final ranking of documents by the degree of relevance and opinionatedness toward a query, previous works linearly combined the opinion and relevance scores without theoretical justification. In [25], Zhang et al. proposed a formal generative model for opinion retrieval that considers the relevant score as a weight for the opinion score of a document. Although their proposed model proved to be effective compared to previous work, it failed to take advantage of the component of the model that aimed to capture topic related opinion expression through proximity. The other shortcoming of their model is that it treats all opinion terms in the lexicon equally, while it is natural to think that some terms are more indicative of an opinion than others.

In this paper we propose a novel probabilistic method that considers the opinionatedness of terms in the lexicon together with its relatedness to the query. We will show the effectiveness of our proposed method in the experimental section.

3. TOPIC RELATED OPINION RETRIEVAL

Blog post opinion retrieval aims at developing an effective retrieval function that ranks blog posts according to the likelihood that they are *expressing an opinion about a particular topic*. We follow the typical generative model in Information Retrieval that estimates the likelihood of generating a document given a query, $p(d|q)$. In opinion retrieval, we also need to estimate the probability of generating an opinion about the query. We introduce the random variable o which denotes the event that the document expresses an opinion about the query. Thus, for opinion retrieval we can rank documents by their likelihood given the query and opinion, $p(d|o,q)$. We then factorize this probability as follows:

$$p(d|o,q) \propto p(d,o,q) \quad = \quad p(d)p(q|d)p(o|q,d) \qquad (1)$$

As was also mentioned in [25], we can see two components in this formula: $p(d)p(q|d)$ which considers the relevance of document to the query, and $p(o|q,d)$ which deals with its "opinionatedness". The relevance probability can be estimated using any existing IR method such as language models [18] or classical probabilistic models [4]. The difference in our model is in the second component, $p(o|q,d)$, that is the opinion score of the document. We propose using a proximity-based estimate as a measure of opinion relatedness to the query.

In the remainder of this section we first explain the non proximity-based method for calculating the opinion score of the document. We then explain our proposed proximity-based opinion scoring.

3.1 Non-Proximity Opinion Score

The first studies on opinion retrieval assumed conditional independence between o and q given the document d. So, $p(o|q,d)$ in those models was calculated as $p(o|d)$. Such models assume that each document discusses only one topic and so if a document is relevant to a query, all opinion expressions in the document are about the query. In order to calculate a non proximity-based (general) opinion score for a document, we can simply calculate the average opinion score over all terms in the document:

$$p(o|q,d) = p(o|d) = \sum_{t \in d} p(o|t)p(t|d) \qquad (2)$$

where $p(t|d) = c(t,d)/|d|$ is the relative frequency of term t in document d and $p(o|t)$ shows the probability of opinionatedness of the term.

3.2 Proximity Opinion Score

The assumption that a document is only relevant to a single topic and that all opinion expressions are about that topic is overly simplistic. In fact, a document can be relevant to multiple topics and just be opinionated about one of them. Therefore, for assigning opinion scores to documents, we need to identify opinion expressions that are directed toward the query topic. One possible approach is to find opinion lexicons that are mostly used to express opinion about the query topic. For instance, the word "delicious" may be used more for expressing opinion about a food type query than an electronic product. Having access to query-related opinion lexicons, we can either ignore the word delicious or give it a low weight if the query is about electronics. For example Na et al. [14] used an opinion lexicon refinement via pseudo relevance feedback in order to build a query-related opinion lexicon.

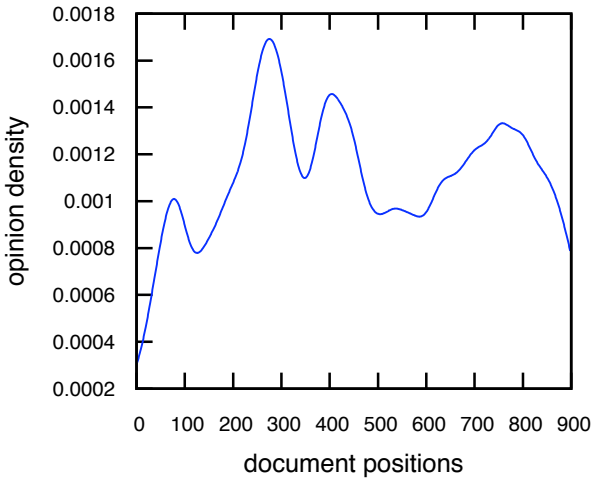

Figure 1: Example of Opinion density at different positions of a document

Another approach is to use the documents' structure. In this approach the distance of an opinion term to the query term is used as a measure of their relatedness. Accordingly, we assume that an opinion term refers with higher probability to the terms closer to its position. On the other hand, opinion terms can refer not only to the entities they proceed or follow, but also to the entities which may be a couple of words, or even sentences, before or after. Bi-gram or tri-gram models have limitations in capturing such dependencies between opinion and topic terms. In order to model this dependency, we propose considering proximity-based density kernels, centered at each opinion term, which favor positions closer to the opinion term's position. As a kernel we can use any non-increasing function of the distance between the position of an opinion term and any other position in a document [10]. We weight this kernel by the probability of opinionatedness of the term. Therefore, the opinion density at each position in the document is the accumulated opinion density from different opinion terms at that position. We define this accumulated probability to be the probability of the opinion expressed in the document about the term at that position. Figure 1 shows the opinion density at different positions in a sample document.

In order to present our model more formally, we first introduce some notation. We denote a document with the vector $d = (t_1, ..., t_i, ..., t_j, ..., t_{|d|})$ where the subscripts i and j indicate positions in the document and t_i indicates the term occurring at the position i. To find the opinion probability at i, we calculate the accumulated opinion probability from all positions of the document at that position. So, for every position j in a document we consider the opinion weight of the term at that position which we denote by $p(o|t_j)$, and we

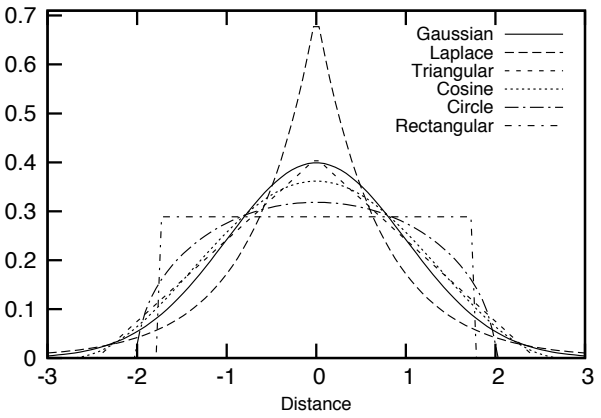

Figure 2: Proximity kernel functions with the same variance.

weight it by the probability that the term at that position j is about the query term at position i. We represent this probability by $P(j|i,d)$ and calculate it as follows:

$$p(j|i,d) = \frac{k(j,i)}{\sum_{j'=1}^{|d|} k(j',i)} \tag{3}$$

here $k(i,j)$, is the kernel function which determines the weight of propagated opinion from t_j to t_i. Thus the probability of opinion at position i in the document can be estimated as:

$$p(o|i,d) = \sum_{j=1}^{|d|} p(o|t_j)p(j|i,d) \tag{4}$$

In the rest of this section we present the different kernels used in our experiments. We investigate the five different density functions used in [10], namely the Gaussian, Triangular, Cosine, Circle and Rectangular kernel. We also present Laplace kernel as an additional kernel in our experiments. Figure 2 shows the different kernels all with the same variance.

In the following formulas, we present normalized kernel functions with their corresponding variance formula.

1. Gaussian Kernel

$$k(i,j) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-(i-j)^2}{2\sigma^2}\right] \tag{5}$$

2. Laplace Kernel

$$k(i,j) = \frac{1}{2b} \exp\left[\frac{-|i-j|}{b}\right] \quad \text{where } \sigma^2 = 2b^2 \tag{6}$$

3. Triangular Kernel

$$k(i,j) = \begin{cases} \frac{1}{a}\left(1 - \frac{|i-j|}{a}\right) & \text{if } |i-j| \le a \\ 0 & \text{otherwise} \end{cases} \quad \text{where } \sigma^2 = \frac{a^2}{6} \tag{7}$$

4. Cosine Kernel

$$k(i,j) = \begin{cases} \frac{1}{2s}\left[1 + cos\left(\frac{|i-j|.\pi}{s}\right)\right] & \text{if } |i-j| \le s \\ 0 & \text{otherwise} \end{cases} \quad \text{where } \sigma^2 = s^2\left(\frac{1}{3} - \frac{2}{\pi^2}\right) \tag{8}$$

5. Circle Kernel

$$k(i,j) = \begin{cases} \frac{2}{\pi r^2}\sqrt{r^2 - (i-j)^2} & \text{if } |i-j| \le r \\ 0 & \text{otherwise} \end{cases} \quad \text{where } \sigma^2 = \frac{r^2}{4} \tag{9}$$

6. Rectangular Kernel

$$k(i,j) = \begin{cases} \frac{1}{2a} & \text{if } |i-j| \le a \\ 0 & \text{otherwise} \end{cases} \quad \text{where } \sigma^2 = \frac{a^2}{3} \tag{10}$$

As one baseline, we compare proximity kernels to the uniform kernel which gives the same importance to all positions in the document and simulates the non proximity-based opinion retrieval presented in section 3.1. Our aim is to investigate whether it is better to use kernels which favor opinion occurrence in close proximity of query term or not.

3.3 Probability of Document Opinionatedness about the Query

Now that we can compute the probability of opinion at each position in the document, we need to calculate an overall probability that the document is expressing an opinion about the query, $p(o|d,q)$. In the following we will suggest different ways for calculating this probability.

$$\begin{aligned} p(o|d,q) &= \sum_{i=1}^{|d|} p(o,i|d,q) \\ &= \sum_{i=1}^{|d|} p(o|i,d,q)p(i|d,q) \end{aligned} \tag{11}$$

We assume that o and q are conditionally independent given the position in the document, $o \perp q|(i,d)$. Thus $p(o|i,d,q)$ reduces to $p(o|i,d)$ which can be estimated using methods proposed in the section 3.2. Here we suggest different methods for estimating $p(i|d,q)$, the probability of position i given the query q and the document. One method assumes that all query terms' positions in the document are equally important. Thus we have:

$$p(i|d,q) = \begin{cases} \frac{1}{|pos(q)|} & \text{if } t_i \in q \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Here $pos(q)$ is the set of all query terms' positions in the document. We can then calculate $p(o|d,q)$ as follows:

$$p(o|d,q) = \frac{1}{|pos(q)|} \sum_{i \in pos(q)} p(o|i,d) \tag{13}$$

As an alternative, we can assume that only the query term position where $p(o|i,d)$ is maximum is important. Thus:

$$p(o|q,d) = max_{i \in pos(q)} p(o|i,d) \tag{14}$$

This approach is similar to first ranking the passages in the document by their degree of opinionatedness and then choosing the most relevant passage for scoring the document. We refer the first and second methods, *avg* and *max* in the experimental section.

We also considered placing a density kernel over each query term position in the document. Since the resulting formula for calculating $p(o|q,d)$ was cubic, the approach was not efficient. It also didn't improved the performance greatly.

3.4 Smoothed Proximity Model

The proximity-based estimate can be further refined by smoothing it with the non proximity-based estimation as follows:

$$p(o|q,d) = (1-\lambda)p(o|q,d) + \lambda p(o|d) \quad (15)$$

Smoothing the proximity model with the non-proximity score lets us capture the proximity at different ranges. This can be useful because there are some documents in which the exact query term occurs rarely. In such documents opinion expressions refer to the query indirectly through anaphoric expressions such as *he, she, it, the film, etc.* Since we don't do any query expansion or reference resolution in our model, we investigate whether smoothing the proximity model with the non-proximity score helps us capture further related opinion expressions in the document.

4. EXPERIMENTAL SETUP

In this section we explain our experimental setup for evaluating the effectiveness of the proposed methods.

Test Collection.

Our experiments are based on the BLOG06 collection and the set of 150 topics for the blog post opinion retrieval task in TREC 2006 through 2008 and their corresponding relevance assessments. The relevance assessments provide information about whether a given blog post is relevant to a topic and also reflects the post opinionatedness nature. Each topic contains three fields of title, description and narrative. We extracted query terms from the title field of a topic. Each permalink component was indexed as a retrieval unit. The preprocessing of the collection was minimal and involved only stopword removal.

In order to be able to compare with TREC 2008 participants, we used 100 topics from TREC 2006 and TREC 2007 numbered 851 to 950 as our training set and 50 topics from TREC 2008, numbered 1001 to 1050, for testing.

Opinion Lexicon.

In our experiments we used the opinion lexicon that was proposed in [9], since it has been shown to be effective in TREC 2008. This lexicon was made using sentiWordNet [3] and an automatically learned model from the Amazon.com product review and specification corpus. The opinion lexicon only contains words from the review corpus that are also present in sentiWordNet (i.e. the intersection of the two word sets). The opinion lexicon model gives us the probability of subjectiveness of each word, $p(sub|w)$, which we use as the probability of opinionatedness (subjectivity) of the word in our model.

Retrieval Baselines.

In order to facilitate direct comparison between systems in TREC 2008 five relevance retrieval baselines were provided by the TREC organizers, selected from the best performing retrieval runs. Each of these baselines covers all 150 topics and contains a list of relevant documents to those topics. Note that the baseline TREC runs are purely relevance retrieval and not opinion retrieval systems. They score relatively well at opinion retrieval simply because the majority of blog posts that are relevant to a topic are also expressing an opinion about it. When evaluating opinion retrieval systems therefore, one must compare the Mean Average Precision (MAP) score of the opinion retrieval system with the MAP (for opinionated posts) of the baseline system. A recent study showed that it is very difficult to improve opinion retrieval performance over a strong baseline on the Blog06 collection[11]. In the following experiments we show the effectiveness of the proposed method in improving the opinion retrieval performance over the best baselines in TREC 2008.

Evaluation.

We used the opinion relevance judgements provided by TREC for evaluation. We report the MAP as well as R-Precision (R-Prec), binary Preference (bPref), and Precision at 10 documents (P@10) in terms of opinion finding.

Throughout our experiments we used the Wilcoxon signed-rank matched pairs test with a confidence level of 0.01 level for testing statistical significance.

5. EXPERIMENTAL RESULTS

In this section we explain the experiments that we conducted in order to evaluate the usefulness of different setting of the proposed method.

5.1 Normalizing Relevance Scores

In order to use the standard TREC baselines in the relevance retrieval component of our model, we need to estimate the probability of relevance of each document to the query. The TREC baselines provide us with relevance score of the document, not the relevance probability. In order to estimate the probability of relevance of a document we investigated different normalization techniques for transforming the relevance score into a probability estimate. The easiest transformation is to use the relevance score directly (rank equivalent to dividing the score by the sum of all document scores for the same query). We also tried normalizing the relevance score using the minimum, $min(score)_q$, maximum $max(score)_q$, mean $mean(score)_q$ and standard deviation $stdev(score)_q$ of document's scores for the query q:

$$N1 = \frac{score - min(score)_q}{max(score)_q - min(score)_q} \quad (16)$$

$$N2 = \frac{score - mean(score)_q}{stdev(score)_q} \quad (17)$$

In addition, we experimented with *Logistic Regression* for learning a transformation from relevance scores to probability estimates. We trained the model using the relevance judgements from the training set. We used variations on the score or rank of the documents in the TREC baselines as a feature for logistic regression. Thus the model we used to estimate the relevance probability of a document in each

	8	16	32	64	128
Score	0.3262	0.3355	0.3364	0.3323	0.3304
N1	0.3641↑	0.3701↑	0.3708↑	0.3685↑	0.3662↑
N2	**0.3713**↑	**0.3732**↑	**0.3740**↑	**0.3720**↑	**0.3709**↑
LRS	0.3205↓	0.33↓	0.3307↓	0.3272↓	0.3254↓
LRLS	0.3216↓	0.3289↓	0.3285↓	0.3248↓	0.3228↓
LRN1	0.331↑	0.3402↑	0.3421↑	0.3395↑	0.3371↑
LRN2	0.2966↓	0.3048↓	0.3055↓	0.3012↓	0.2969↓
LRR	0.3324	0.33	0.3291	0.3297	0.3302
LRLR	0.3613↑	0.3672↑	0.3688↑	0.3682↑	0.3673↑

Table 1: MAP over TREC baseline4 using laplace kernel with different sigma values. Rows show MAP using different relevant probability estimation methods. An uparrow(↑) and downarrow(↓) indicate statistically significant increase and decrease over using the score directly.

baseline was the following:

$$p(d \text{ is relevant} | TREC\_baseline_j) = \frac{e^{\alpha+\beta.x}}{1+e^{\alpha+\beta.x}} \quad (18)$$

Where x is one of the normalized scores, the rank or the log of the rank (or score) of document d. In order to estimate this probability, we learn values for α and β. We used the logistic regression implementation provided in LingPipe[1], the TREC 2006 topics, and the set of relevant and non-relevant documents for learning these parameters.

Table 1 shows the opinion retrieval performance of our proposed system, using different probability estimation methods on TREC baseline 4. We report the results for exponentially increasing values of sigma. In this table LRS, LRLS, LRN1 and LRN2, LRR and LRLR denote logistic regression using the score, log of the score, normalized score using equation 16 and 17, using rank of documents instead of score and log of rank as the explanatory variable respectively. As can be seen form Table 1, N1, N2 and LRLR have the highest MAP over all sigma values on TREC baseline 4 and the improvement over the score is statistically significant, but there is no statistically significant difference between these three methods. We chose N2 for TREC baseline 4 as it had the highest MAP over training topics. For the other baselines, we found that LRLR performed best on baseline 1,3 and 5, and N2 on baseline 2.

5.2 Probability of Document Opinionatedness about the Query

In section 3.3 we proposed two different techniques for calculating the relevant opinion probability. Table 2 shows the result of using *avg* and *max* techniques for different sigma values. The results show that the *max* is statistically better than the *avg* method. Therefore, we use the *max* method in the rest of our experiments.

5.3 Parameter Selection for the Proximity-based Relevant Opinion Model

In section 3.2 we proposed a proximity-based opinion propagation method in which a proximity kernel is considered around each opinion term occurrence position in the document. The opinion density at a query term position is then calculated by counting the accumulated opinion density from different opinion terms at that position. In this way, a query term which occurs at a position close to many opinionated terms will receive high opinion density.

[1]http://alias-i.com/lingpipe/

	8	16	32	64	128
avg	0.3526	0.3573	0.3617	0.3642	0.3668
max	**0.3713**∗	**0.3732**∗	**0.3740**∗	**0.3720**∗	**0.3709**∗

Table 2: MAP over TREC baseline4 with different opinion scoring method for laplace kernel with different sigma values. A star(∗) indicates statistically significant improvement over *avg* method

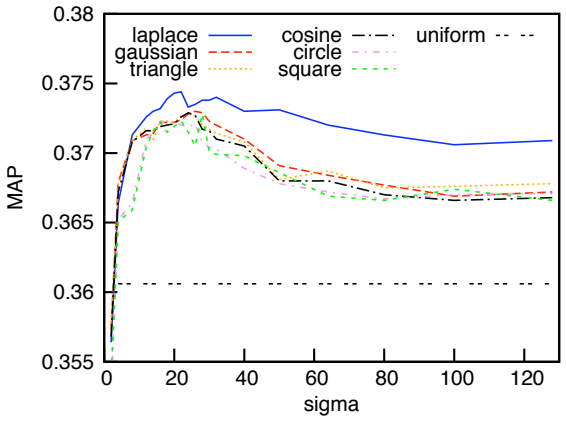

Figure 3: Parameter sweep for the kernel functions

The proposed relevant opinion model has two parameters: the type of kernel function and its bandwidth parameter σ which adjusts the scope of opinion propagation (referencing) over the document. Performance of different kernels on training topics using the best parameter for each kernel is reported in Table 3. The result shows that all proximity kernels improve significantly over the non-proximity baseline, but there is no statistically significant difference between different proximity kernels when using the best parameters for each kernel. Fig. 3 reports the sensitivity (in terms of MAP) of the different kernels to different values of σ parameters ranging from 2 to 128. Although there was no statistically significant difference between kernels, the Laplace kernel has the most effective and stable MAP over different parameter settings. Thus, we used it as the proximity kernel for our system evaluation on the test query set.

kernel	σ	MAP	R-prec	bPref	p@10
Laplace	22	**0.3744**∗	**0.4113**	**0.4305**	**0.6200**
Gaussian	26	0.3730∗	0.4099	0.4305∗	0.6160
Cosine	24	0.3729∗	0.4095	0.4305	0.6170
Triangle	24	0.3728∗	0.4086	0.4302	0.6180
Square	28	0.3728∗	0.4100	0.4300	0.6130
Circle	16	0.3723∗	0.4080	0.4298	0.6120
Uniform	∞	0.3606	0.4011	0.4231	0.6190

Table 3: The performance of proximity-based opinion retrieval for the best σ for each kernel. ∗ indicates statistically significant improvement over constant kernel.

kernel	λ	σ	MAP	R-prec	bPref	p@10
Laplace	0.4	12	0.3775	0.4166	0.4325	0.6400
Gaussian	0.6	4	0.3772	0.4147	0.4317	0.6360
Triangle	0.5	4	0.3764	0.4121	0.4317	0.6420
Cosine	0.6	4	0.3762	0.4142	0.4318	0.6390
Circle	0.7	4	0.3764	0.4167	0.4333	0.6360
Square	0.4	18	0.3757	0.4092	0.4326	0.6230

Table 4: The performance of Opinion Mixture model for the best σ and λ for each kernel.

5.4 Parameter Selection for the Smoothed Proximity Model

We now report our experimental results in finding the best parameters for the smoothed proximity model presented in section 3.4. This model has three parameters: kernel type, σ and the λ parameter which is the interpolation coefficient. In order to find the best parameters, we tried different λ values for each σ value in the range of [2,128]. Table 4 reports the performance of different kernels using the best σ and λ parameter pairs for each kernel. It shows that, interpolating the proximity score with the no proximity opinion score improves the performance. It also shows that there is no statistically significant difference between kernels when the proximity score is interpolated with the general opinion score of the document. Here again, we choose the Laplace kernel for the evaluation on the test set.

5.5 Experimental Results on the Test Query Set

In this section, we present the evaluation results of our approaches on the TREC 2008 query topics. Table 5 presents the retrieval performances of the proposed methods over the five standard TREC baselines in terms of opinion retrieval. We also compare the performance of the proposed techniques to the no proximity opinion retrieval method. In Table 5, we show results for the non-proximity method (noprox), the laplace kernel (laplace) and the smoothed model using the laplace kernel (laplaceInt). Table 5 shows that the proposed methods are consistently effective across all five standard TREC baselines.

We also performed per topic analysis of performance for 2008 topics. The results showed that the proposed opinion retrieval methods improves Average Precision of 40 out of 50 test topics over TREC baseline 4. Topics which achieved the highest improvement over the baseline are mostly single word topics such as topic 1001 (Carmax) and topic 1023 (Yojimbo). They performed poorly on topics with multiple terms such as topic 1013 (Iceland European Union). The reason is that topics with multiple terms usually define a concept which is different from each single term in the query. Thus a more precise model for capturing the occurrence of all query terms and their proximity is required to handle such queries.

Finally we compare our proposed approaches with the best runs at TREC 2008 blog track and report the comparison result in table 6 and table 7. Table 6 shows the performance of our proposed methods on the standard TREC baseline4, comparing to the best TREC run and the later proposed method in [22](KLD+dist-FD-FV-subj-b4) on the same baseline. Interestingly, both proposed methods outperform the best reported results in TREC (B4PsgOpinAZN)

	MAP	R-prec	bPref	p@10
basline1	0.3239	0.3682	0.3514	0.5800
noprox	0.3751$*$	0.4154$*$	0.4082	0.6720$*$
laplace	0.3960$*\dagger$	0.4369$*$	0.4291$*\dagger$	0.6860$*$
laplaceInt	**0.4020**$*\dagger$	**0.4412**$*\dagger$	**0.4326**$*\dagger$	**0.6920**$*$
baseline2	0.2639	0.3145	0.2902	0.5500
noprox	0.2791$*$	0.3299	0.3066	0.5740
laplace	0.2881$*$	0.3401$*$	0.3166$*$ $\dagger$	0.5820
laplaceInt	**0.2886**$*\dagger$	**0.3411**$*$	**0.3166**$*\dagger$	**0.5860**
basline3	0.3564	0.3887	0.3677	0.5540
noprox	0.3819	0.4188$*$	0.4075	0.6400$*$
laplace	0.3989$*\dagger$	0.4369$*$	0.4207$*$	0.6600$*$
laplaceInt	**0.4043**$*\dagger$	**0.4389**$*\dagger$	**0.4247**$*$	**0.6660**$*$
baseline4	0.3822	0.4284	0.4112	0.6160
noprox	0.4129$*$	0.4460$*$	0.4368$*$	0.6880
laplace	0.4267$*$	0.4545$*$	0.4472$*$	0.7080$*$
laplaceInt	**0.4292**$*\dagger$	**0.4578**$*$	**0.4485**$*$	**0.7140**$*$
baseline5	0.2988	0.3524	0.3395	0.5300
noprox	0.2918	0.3455	0.3497	0.5980
laplace	0.3188$*\dagger$	0.3732$\dagger$	0.3698$*\dagger$	0.6080$*$
laplaceInt	**0.3223**$*\dagger$	**0.3785**$\dagger$	**0.3715**$*\dagger$	**0.6120**$*$

Table 5: Opinion finding MAP results over five standard TREC baselines using different proximity methods for TREC 2008 topics. A star($*$) and dagger($\dagger$) indicate statistically significant improvement over the relevance and non-proximity opinion retrieval baselines respectively.

Run	Map	Δ MAP
laplaceInt	0.4292	12.30%
laplace	0.4267	11.64%
KLD+dist-FD-FV-subj-b4	0.4229	10.65%
B4PsgOpinAZN	0.4189	9.60%

Table 6: Opinion finding results for best runs on standard baseline 4, ranked by Mean Δ MAP using TREC 2008 new topics

and KLD+dist-FD-FV-subj-b4. B4PsgOpinAZN is based on a query specific lexicon which is built via feedback-style learning. KLD+dist-FD-FV-subj-b4 uses wikipedia for finding different facets in the query and query expansion. It then used Kullback-Leibler divergence to weight subjective units occurring near query terms. The distance of the query term to the subjective units is also considered in this model.

Table 7 reports the mean MAP and the mean of their relative improvements over the five standard baselines (ΔMAP). We observe that the proposed methods have the highest mean of MAP and mean of ΔMAP across the five standard baselines. This indicates that the proposed methods are effective and stable across different relevance retrieval baselines.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel probabilistic model for blog opinion retrieval. We focused on the problem of opinion topic relatedness and we showed that using proximity information of opinionated terms to query terms is a good indicator of opinion and query-relatedness. We studied the parameter selection for our model and we have shown that when kernels are compared using the best parameter for

Run	Map		Δ MAP	
	Mean	stdev	Mean	stdev
laplaceInt	0.3693	0.06	13.41%	6.38%
laplace	0.3657	0.06	12.33%	5.94%
uicop1bl1r	0.3614	0.04	11.76%	6.93%
B1PsgOpinAZN	0.3565	0.05	9.67%	0.77%

Table 7: Opinion finding results for best runs using all five standard baselines, ranked by Mean Δ MAP using TREC 2008 new topics

each, there is no statistically significant difference between them. We proposed using Laplace kernel as it was more stable on different parameter values. We also analyzed the effect of normalizing the relevance score before applying it in the model. Our results show that normalization can be important, and that the best normalization strategy is dependent on the underling relevance retrieval baseline.

We have evaluated the proposed method on the BLOG06 collection. The proposed model was shown to be effective across five standard relevance retrieval baselines. It achieved the highest improvement over the best standard TREC baseline (baseline 4), comparing to other reported results on the same baseline.

For future work we plan to investigate the effect of using reference resolution techniques on the performance of the proposed method.

7. ACKNOWLEDGMENTS

We thank Seung-Hoon Na from the KLE group of Pohang University of Science and Technology for providing the opinion lexicon. This research was partly funded by the "Secrétariat d'état à l'Éducation et à la Recherche (SER)" and COST Action IC0702 "Combining Soft Computing Techniques and Statistical Methods to Improve Data Analysis Solutions".

8. REFERENCES

[1] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of WWW '03*, pages 519–528, 2003.

[2] K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of EMNLP'06*, pages 345–354, 2006.

[3] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC '06*, pages 417–422, 2006.

[4] N. Fuhr. Probabilistic models in information retrieval. *Proceedings of Comput. J.*, 35(3):243–255, 1992.

[5] M. Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *COLING '04*, page 841, 2004.

[6] B. He, C. Macdonald, I. Ounis, J. Peng, and R. L. Santos. University of glasgow at TREC 2008: Experiments in blog, enterprise, and relevance feedback tracks with terrier. In *Proceedings of TREC'08*, 2008.

[7] M. Hurst and K. Nigam. Retrieving topical sentiments from online document collections. In *Document Recognition and Retrieval XI*, pages 27–34, 2004.

[8] L. Jia, C. T. Yu, and W. Zhang. UIC at TREC 2008 blog track. In *Proceedings of TREC'08*.

[9] Y. Lee, S.-H. Na, J. Kim, S.-H. Nam, H.-Y. Jung, and J.-H. Lee. KLE at TREC 2008 blog track: Blog post and feed retrieval. In *Proceedings of TREC'08*, 2008.

[10] Y. Lv and C. Zhai. Positional language models for information retrieval. In *SIGIR '09*, pages 299–306, 2009.

[11] C. Macdonald, B. He, I. Ounis, and I. Soboroff. Limits of opinion-finding baseline systems. In *SIGIR '08*, pages 747–748, 2008.

[12] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC-2007 blog track. In *Proceedings of TREC'07*, 2007.

[13] T. Mullen and N. Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP'04*, pages 412–418, 2004.

[14] S.-H. Na, Y. Lee, S.-H. Nam, and J.-H. Lee. Improving opinion retrieval based on query-specific sentiment lexicon. In *ECIR '09*, pages 734–738, 2009.

[15] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the TREC-2006 blog track. In *Proceedings of TREC'06*, 2006.

[16] I. Ounis, C. Macdonald, and I. Soboroff. Overview of the TREC-2008 blog track. In *Proceedings of TREC'08*, 2008.

[17] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of EMNLP '02*, pages 79–86, 2002.

[18] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR '98*, pages 275–281, 1998.

[19] R. L. Santos, B. He, C. Macdonald, and I. Ounis. Integrating proximity to subjective sentences for blog opinion retrieval. In *Proceedings of ECIR'09*, pages 325–336, 2003.

[20] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02*, pages 417–424, 2002.

[21] P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346, 2003.

[22] O. Vechtomova. Facet-based opinion retrieval from blogs. *Inf. Process. Manage.*, 46(1):71–88, 2010.

[23] K. Yang. WIDIT in TREC 2008 blog track: Leveraging multiple sources of opinion evidence. In *Proceedings of TREC'08*, 2008.

[24] 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*, page 427, 2003.

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

[26] W. Zhang, C. Yu, and W. Meng. Opinion retrieval from blogs. In *CIKM '07*, pages 831–840, 2007.