Fusion Approach to Finding Opinions in Blogosphere

Kiduk Yang
Indiana University
1320 E. 10th Street
Bloomington, IN 47405
1-812-855-2793
kiyang@indiana.edu

Ning Yu
Indiana University
1320 E. 10th Street
Bloomington, IN 47405
1-812-856-5874
nyu@indiana.edu

Alejandro Valerio
Indiana University
150 S.Woodlawn Ave.
Bloomington, IN 47405
1-812-855-6486
avalerio@indiana.edu

Hui Zhang
Indiana University
1320 E. 10th Street
Bloomington, IN 47405
1-812-856-5874
hz3@indiana.edu

Weimao Ke
Indiana University
1320 E. 10th Street
Bloomington, IN 47405
1-812-856-5874
wke@indiana.edu

Abstract
In this paper, we describe a fusion approach to finding opinion about a given target in blog postings. We tackled the opinion blog retrieval task by breaking it down to two sequential subtasks: on-topic retrieval followed by opinion classification. Our opinion retrieval approach was to first apply traditional IR methods to retrieve on-topic blogs, and then boost the ranks of opinionated blogs using combined opinion scores generated by four opinion assessment methods. Our opinion module consists of Opinion Term Module, which identify opinions based on the frequency of opinion terms (i.e., terms that only occur frequently in opinion blogs), Rare Term Module, which uses uncommon/rare terms (e.g., "sooo good") for opinion classification, IU Module, which uses IU (I and you) collocations, and Adjective-Verb Module, which uses computational linguistics' distribution similarity approach to learn the subjective language from training data.

General Terms
Algorithms, Performance, Experimentation

Keywords
Method Fusion, Rank-boosting, Opinion Identification, Dynamic Tuning

1. Introduction
Blogs, journal-like Web pages that started out as online diaries about a decade ago, has evolved in recent years to become one of the mainstream tools for collaborative content creation on the Web. Perhaps the most distinguishing characteristic of the blogs is their highly personalized nature, often containing personal feelings, perspectives, and/or opinions about a topic. Text Retrieval Conference (TREC), an international information research forum that supports a variety of cutting-edge information retrieval research, explored the question of how to find such "opinionated" blog entries in one of its specialized venue called the *blog track*. The Web Information Discovery Integrated Tool (WIDIT) laboratory of Indiana University, which researches various fusion approaches to knowledge discovery, participated in the blog track of TREC-2006. This paper describes the WIDIT's fusion approach that combines multiple methods of finding opinion blogs.

The blog opinion retrieval task, as described by TREC, is to "uncover the public sentiment towards a given entity/target". TREC blog topics,[1] as is the case with a typical Web query, are very brief (e.g. "skype"), so retrieving blogs about a given topic is not trivial, let alone finding blogs that express opinions on the topic. For example, a post with "skype me at username ..." is obviously not related to the topic "skype". Even if topically relevant blogs were to be retrieved, identifying opinionated posts among them is quite a challenging task.

In short, blog opinion retrieval faces two main challenges: to find blogs about a topic and to identify opinionated ones among them. The difficulty with on-topic retrieval stems from the shortness of the query that causes retrieval of topically non-relevant blogs containing the query terms, whereas the difficulty with opinion identification is due to the context-dependent nature of subjective language. According to Wiebe et al. [17], "both opinionated and factual documents tend to be composed of a mixture of subjective and objective language." In other words, it is hard to differentiate opinionated documents from factual ones with simple clues.

To figure out possible solutions, we review research on evaluating subjective messages in Section 2.

2. Related Work
Efron [5] presents an interesting hyperlink-based approach for identifying subjective affiliations of Web documents (e.g., blogs) and estimating political orientation of those documents. The proposed model estimates the likelihood of co-citation between a document (which orientation is yet to be known) and documents of known orientations. It is still arguable whether hyperlinks are good indicators of subjective affiliations. Nevertheless, given the results by Herring et al. [8] that blog links are selective, we feel more confident that connections convey such affiliation information. The question is, for blogs commenting certain products, whether they are more likely to have links to product descriptions (facts) or to other opinionated pages.

By using text analysis and external knowledge (i.e. Amazon's Web Services for locating products), Mishne and de Rijke [12] presents a method for analyzing blogs and deriving product wish lists. Specifically, the method tries to identify references to books by recognizing proximities of keywords (e.g., "read", "book") and relevant patterns (e.g., [ENTITY] by [PERSON]). It employs a keyword extraction method to retrieve words that appear more frequently on one blog than others. We can also figure out other words (such as "try", "product", etc) and patterns (such as "released by [ENTITY]") that are used frequently for

[1] "Topic" is a TREC terminology that refers to a statement of information need.

commenting purposes. Thus, this method is applicable to identifying posts that is about something and possibly commenting on that.

Following a similar research direction, Liu et al. [11] propose a framework for analyzing and comparing customer reviews of products and a technique based on language pattern mining to extract product features from Pro and Cons. The authors have implemented a system called Opinion Server that integrates visualization methods to present retrieved results and compare customer reviews. Given three major review formats, the paper focuses on format, in which reviewers describe Pro and Cons in details separately. This method is hardly useful because blog posts are rarely well formatted.

In a more practical research, Hu and Liu [9] examine the problem of generating feature-based summaries of customer reviews of products sold online. Given a set of customer reviews of a particular product, authors divide the process of generating summaries into three subtasks. 1) identify features of the product that customers have expressed their opinions on; 2) identify review sentences that give positive or negative opinions for each feature; and 3) produce a summary using the discovered information. Association mining is used to find frequent noun/noun phrases which are likely to be product features. Instead of classifying each review as a whole, this research classifies each sentence in a review and identifies orientation of the sentence (negative or positive). This paper uses adjectives as opinion words to identify opinion sentences. Part-of-speech tagging from natural language processing is used to find opinion features and opinion words. The research proposes a simple method to utilize the adjective synonym/antonym set in WordNet and predict the semantic orientations of adjectives. This seems to be a very effective method with 0.8 average sentence orientation accuracy.

The presented method of opinion word identification using adjectives will be useful. However, this is not sufficient in the cases when opinions are expressed with adverbs, verbs and nouns. Opinion word orientation identification using WordNet synsets technique is a reasonable way to expand initial seeds. This technique can be extended to identify subjectivity and/or subjective orientations of blog posts.

Chklovski does a similar research. His paper [4] focuses on automatic summarization of opinions and assessments stated on the web in product reviews, discussion forums, and blogs. It presents a system called GrainPile for this purpose, which recognizes subjective expressions (e.g., fairly, very, extremely) and maps/aggregates them to a common scale. Results show that this approach strongly outperforms an interpretation-free and co-occurrence based method. Although the paper aims to quantify the degree of opinions, the method for identifying subjective adverbs and adverbial phrases such as ”fairly”, ”very”, ”not too”, ”pretty darn” could be used to tackle the problem of identifying opinionated posts as well.

For subjectivity recognition, much research has been done in the field of Natural Language Processing. Wiebe et al. [17] and Wilson et al. [18] introduce theoretical background and practical methods used for learning subjective language from text corpora, involving information extraction and text categorization. It presents several categories of subjectivity clues, which include low-frequency words, n-gram collocations, and adjectives and verbs. Then these clues are evaluated and used together to perform opinion piece recognition tasks.

Opinion piece recognition is essentially a binary text categorization task–a piece belongs to either “opinionated messages” or “non-opinionated messages”. Although some subjective clues need training, others such as unique/low-frequency terms and n-gram with unique/low-frequency term patterns do not have this requirement and prove to be effective. Although identifying distributional similarities requires a document collection to be pre-analyzed, it does not require document labels, i.e., “subjective” or “objective” annotations. Therefore, these approaches can be implemented without training data. Concerning that subjective clues are context-dependent, the paper proposes a method to measure Subjectivity Density by taking into account proximity of potentially subjective elements.

3. Research Question

Having developed its own topical search system over the years, WIDIT focused on the question of how to adapt a topical retrieval system for opinion retrieval task. The intuitive answer was to first apply existing system to retrieve blogs about a target (i.e., on-topic retrieval), optimize on-topic retrieval to address the challenges of short queries, and then identify opinion blogs by leveraging evidences of subjectiveness/opinion (i.e., opinion identification).

Two key research questions at this point are how to optimize on-topic retrieval, and a compound question of what the evidences of opinion are and how they can be leveraged to retrieve opinionated blogs. As for the opinion identification, we considered the following three sources of evidence:

- Opinion Lexicon: One obvious source of opinion is a set of terms often used in expressing opinions (e.g., “Skype *sucks*”, “Skype *rocks*”, “Skype is *cool*”).
- Opinion Collocations: One of the contextual evidence of opinion comes from collocations used to mark adjacent statements as opinions (e.g., “*I believe* God exists”, “God is dead *to me*”).
- Opinion Morphology: When expressing strong opinions or perspectives, people often use morphed word form for emphasis (“Skype is *soooo* buggy”, “Skype is *bugfested*”).

4. Methodology

As described in the preceding section, our approach consists of three main steps: initial retrieval, on-topic retrieval optimization, and opinion identification. Initial retrieval is executed using the standard WIDIT retrieval method, on-topic retrieval optimization is done by a post-retrieval reranking approach that leverages multiple topic-related factors, and opinion identification is accomplished by a fusion of four opinion modules that leverages multiple sources of opinion evidence. The overview of WIDIT blog opinion retrieval system is shown in Figure 1.

4.1 Initial Retrieval

4.1.1 Indexing

The initial retrieval is executed by the WIDIT retrieval engine, which consists of document/query indexing and retrieval module. After removing markup tags and stopwords, WIDIT’s indexing modules applies a modified version of the simple plural remover

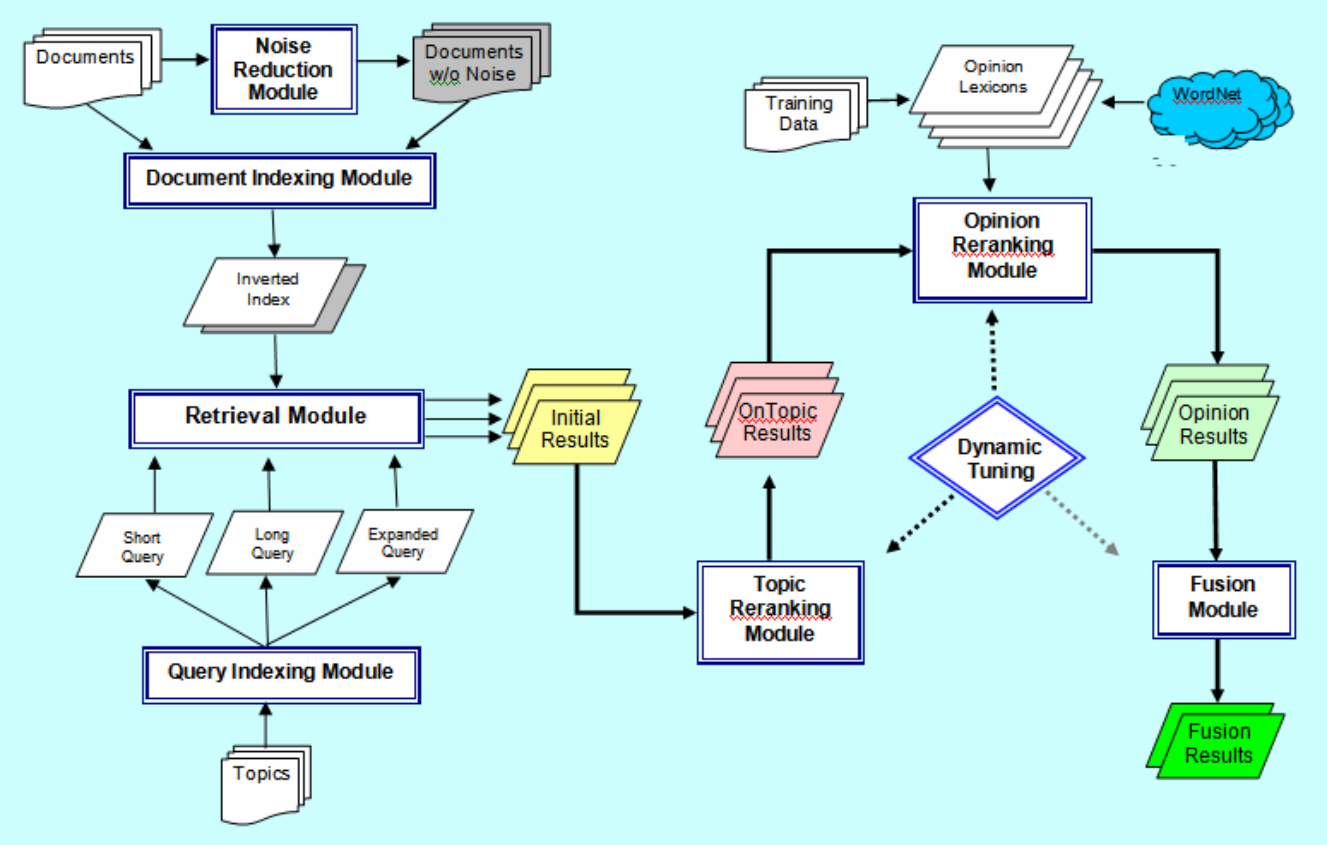

Figure 1: *WIDIT Blog Opinion Retrieval System Architecture*

[7].[2] The stopwords consisted of non-meaningful words such as words in a standard stopword list, non-alphabetical words, words consisting of more than 25 or less than 3 characters, and words that contain 3 or more repeated characters. Hyphenated words were split into parts before applying the stopword exclusion, and acronyms and abbreviations were kept as index terms[3].

In order to enable incremental indexing as well as to scale up to large collections, WIDIT indexes the document collection in fixed-size subcollections, which are searched in parallel. The whole collection term statistics, derived after the creation of the subcollections, are used in subcollection retrievals so that subcollection retrieval results can simply be merged without any need for retrieval score normalizations.

Query indexing module include query expansion submodules that identify nouns and noun phrases, expand acronyms and abbreviations, and extract non-relevant portion of topic descriptions with which to formulate various expanded versions of the query.

4.1.2 Retrieval

The retrieval module implements both Vector Space Model (VSM) using the SMART length-normalized term weights and the probabilistic model using the Okapi BM25 formula. For the VSM implementation, SMART *Lnu* weights with the slope of 0.3 are used for document terms [3], and SMART *ltc* weights [2] are used for query terms. *Lnu* weights attempt to match the probability of retrieval given a document length with the probability of relevance given that length [15].

$$d_{ik} = \frac{\log(f_{ik})+1}{\sqrt{\sum_{j=1}^{t}(\log(f_{ij})+1)^2}} \qquad q_k = \frac{(\log(f_k)+1)*idf_k}{\sqrt{\sum_{j=1}^{t}[(\log(f_j)+1)*idf_j]^2}} \quad (1)$$

Equation (1) describes the SMART formula, where d_{ik} is the document term weight (*Lnu*), q_k is the query term weight (*ltc*), f_{ik} is the number of times term k appears in document i, f_k is the number of times term k appears in the query, idf_k is the inverse document frequency of term k, and t is the number of terms in document or query.

$$d_{ik} = \log\left(\frac{N - df_k + 0.5}{df_k + 0.5}\right)\frac{f_{ik}}{k_1((1-b)+b\cdot(\frac{dl}{avdl}))+f_{ik}} \quad (2)$$

$$q_k = \frac{(k_3+1)f_k}{k_3+f_k}$$

[2] The simple plural remover was chosen to speed up indexing time and to minimize the overstemming effect of more aggressive stemmers.

[3] Acronym and abbreviation identification was based on simple pattern matching of punctuations and capitalizations.

The simplified version of the Okapi BM25 relevance scoring formula [13], which is used to implement the probabilistic model, is described in equation (2), where *N* is the number of documents in the collection, *df* is the document frequency, *dl* is the document length, *avdl* is the average document length, and k1, b, k3 are parameters (1.2, 0.75, 7 to 1000, respectively).

4.2 On-Topic Retrieval Optimization

In order to optimize topical retrieval performance in top ranks, the initial retrieval results are reranked based on a set of topic-related reranking factors. The topic reranking factors used are *Exact Match*, which is the frequency of exact query string occurrence in document normalized by document length, *Proximity Match*, which is the length-normalized frequency of padded[4] query string occurrence, *Noun Phrase Match*, which is the length-normalized frequency of query noun phrases occurrence, and *Non-Rel Match*,[5] which is the length-normalized frequency of non-relevant nouns and noun phrase occurrence. The on-topic reranking method consists of following three steps:

(1) Compute topic reranking scores for each of top N results.

(2) Categorize the top N results into reranking groups designed to preserve initial ranking while appropriate rank-boosting for a given combination of reranking factors.

(3) Boost the rank of documents using reranking scores within groups.

The objective of reranking is to float low ranking relevant documents to the top ranks based on post-retrieval analysis of reranking factors. Although reranking does not retrieve any new relevant documents (i.e. no recall improvement), it can produce high precision improvement via post-retrieval compensation (e.g. phrase matching).

4.3 Opinion Identification

Opinion identification is accomplished by combining the four opinion modules that leverage various evidences of opinion (e.g, Opinion Lexicon, Opinion Collocation, Opinion Morphology). The modules are *Opinion Term Module*, which identify opinions based on the frequency of opinion terms (i.e., terms that only occur frequently in opinion blogs), *Rare Term Module*, which uses uncommon/rare terms (e.g., "sooo good") for opinion classification, *IU Module*, which uses IU (I and you) collocations, and *Adjective-Verb Module*, which uses computational linguistics' distribution similarity approach to learn the subjective language from training data. Opinion modules require opinion lexicons, which are extracted from training data. We constructed 20 training topics from BlogPulse (http://www.blogpulse.com/) and Technorati search (http://www.technorati.com/) archives and manually evaluated the search results of the training topics to generate the training data set of 700 blogs.

The application of opinion modules is similar to on-topic retrieval optimization in that opinion scores generated by modules act as opinion reranking factors to boost the ranks of opinionated blogs in the topic-reranked results.

[4] "Padded" query string is a query string with up to *k* number of words in between query words.

[5] Non-rel Match is used to suppress document instead of boosting.

4.3.1 Opinion Term Module

The basic idea behind the *Opinion Term Module* (OTM) is to identify opinion blogs based on the frequency of opinion terms, which are terms that only occur frequently in opinion blogs. OTM computes opinion score using an OT lexicon, which we created by extracting terms from positive training data using information gain, excluding terms appearing in negative training data, and manually selecting a set of opinion terms. Two OTM scores are generated: document-length normalized frequency of OT terms in document and OT terms near query string in document.

4.3.2 Rare Term Module

Rare Term Module (RTM) is derived from the hypothesis that people become creative when expressing opinions and tend to use uncommon/rare terms (e.g., "sooo good"). Thus, we extracted low frequency terms from positive training data, removed dictionary terms, and examined them to construct a RT lexicon and regular expressions that will identify creative term patterns used in opinion blogs. Two RT scores similar to OT scores are computed.

4.3.3 IU Module

IU Module (IUM) is based on the observation that pronouns such as 'I' and 'you' appear very frequently in opinion blogs. For IU lexicon construction, we compiled a list of IU (I and you) collocations from training data (e.g., 'I believe', 'my assessment', 'good for you', etc.). IUM counts the frequency of "padded" IU collocations within sentence boundary to compute two IUM scores similar to OTM and RTM.

4.3.4 Adjective-Verb Module

The hypothesis underlying Adjective-Verb module (AVM) is similar to OTM in that it assumes high frequency of opinion terms in opinion blogs. In addition to restricting opinion terms to verbs and adjectives, AVM differs from OTM in its lexicon construction by using computational linguistics' distribution similarity approach that attempts to learn the subjective language from training data rather than shallow linguistic approaches of other opinion modules.

The Adjective/Verb component uses the density of potentially subjective elements (PSE) to determine the subjectivity of blog posts. It assumes that a post with a high concentration of subjective adjectives and verbs must be opinionated. These parts of speech are the ones that better reveal the author's intention by using attributes ("good", "bad", "ugly") or expressing reactions to ideas or objects ("hate", "love", "disgust"). The idea was evaluated by Wiebe et al. [17] with successful results and their algorithm was the starting point for the design of the component.

The component relies heavily on the elements of the PSE set, so their selection is a key process that must be done carefully. Ideally, the PSE set should be broad so that the wide variety of terms used to describe opinion is captured, but at the same time should not include ambiguous terms that may lead to false positives. For this purpose, an initial PSE set of subjective terms is manually collected. The seed set is then expanded, first by gathering related terms from several lexical references, and second by finding terms that co-occur with PSEs in opinionated posts. Next, the set of candidate PSE is refined by verifying its classification performance against a validation set and removing

the elements that lead to misclassifications. The PSE set is cleaned up manually at the end of the process and also at several points between the execution steps.

4.3.4.1 Selection of Potential Subjective Elements

The collection of PSEs is executed in two main steps: (1) Expansion of an initial seed set and (2) Refinement of the candidate set to eliminate ambiguous elements. The initial seed set of adjectives and verbs is collected manually, including words such as "good", "bad", "oppose", and "agree".

The expansion of the initial seed set is done by looking up the terms into several lexical references, namely WordNet, Levin's verb class, and FrameNet. Next, the related adjectives and verbs are searched using a specific process for each reference, and the new words are added to the set. Specifically for WordNet, each word in the current PSE set is searched and its synonyms, "related terms", and antonyms are added to the set. The process is iterated for each new term until no more new terms are found. The antonym relation is particularly useful to expand the set since it gathers adjectives with opposite orientations. For example, the term "slow" can be found from the term "fast". This proved effective for finding opinionated texts in [9]. To speed up the following stage, the PSE set undergoes a preliminary clean up that removes all terms that appear less than 1000 times in all the Blog-TREC collection. A manual filtering is also executed.

The next step of PSE expansion uses distributional similarity to identify new adjectives and verbs from a pre-classified set of opinionated blog posts. In this technique, presented in [17], the co-occurrence of words in a text indicate that the words share some practical use, as when expressing opinion, therefore are considered similar. An important observation is that low-frequency words are found often in opinionated texts and may be difficult to judge its importance. So, to decide whether to select a word as a PSE, the word is not considered individually, but together with a cluster of words similar to it. Using the distributional similarity algorithm, the words that are found similar to the ones on the PSE set can be added to the set.

Finally, the resulting expanded PSE set is cleaned up by the last step of the distributional similarity algorithm, which checks the candidate PSE subsets using the validation-data (typically the half of the training data not already used). Through this step it is possible to remove PSEs from the seed set that affect the classification performance. A final manual cleanup is also executed.

4.3.4.2 Classifying Blogs using AVM

A blog post is classified as opinionated or non-opinionated based on the density of PSEs in its content. The density is defined as the proportion of all adjectives and verbs in the post that are PSEs. Two threshold values were defined: T_1 indicates the lowest PSE density that a post may have for it to be considered opinioned, so that any post with a PSE density value of T_1 or higher is classified as opinionated with 100% confidence. Analogously, T_2 is the high limit for a post to be classified as non-opinionated with 100% confidence. If the PSE density D of a post is between the two thresholds, the confidence of the result is proportional to the distance of D to T_1 and T_2. The values for T_1 and T_2 that produced the best results were 0.5 and 0.2 respectively.

4.4 Fusion

The fusion module combines the multiple sets of search results after retrieval time. In addition to two of the most common fusion formulas, *Similarity Merge* [6, 7] and *Weighted Sum* [1, 15], WIDIT employs variations of the weighted sum formula. The similarity merge formula multiplies the sum of fusion component scores for a document by the number of fusion components that retrieved the document (i.e. overlap), based on the assumption that documents with higher overlap are more likely to be relevant. Instead of relying on overlap, the weighted sum formula sums fusion component scores weighted with the relative contributions of the fusion components that retrieved them, which is typically estimated based on training data. Both formulas compute the fusion score of a document by a linear combination of fusion component scores.

$$FS_{WS} = \sum(w_i * NS_i), \quad (3)$$

$$FS_{OWS} = \sum(w_i * NS_i * olp), \quad (4)$$

$$FS_{WOWS} = \sum(w_i * NS_i * w_i * olp), \quad (5)$$

where:

FS = fusion score of a document,
w_i = weight of system i,
NS_i = normalized score of a document by system i,
$= (S_i - S_{min}) / (S_{max} - S_{min})$
olp = # of systems that retrieve a given document.

In our earlier study [20], similarity merge approach proved ineffective when combining content- and link-based results, so we devised three variations of the weighted sum fusion formula, which were shown to be more effective in combining fusion components that are dissimilar [19]. Equation (3) describes the simple *Weight Sum* (WS) formula, which sums the normalized system scores multiplied by system contribution weights. Equation (4) describes the *Overlap Weight Sum* (OWS) formula, which multiplies the WS score by overlap. Equation (5) describes the *Weighted Overlap Weighted Sum* (WOWS) formula, which multiplies the WS score by overlap weighted by system contributions. The normalized document score, NS_i, is computed by Lee's min-max formula [10], where S_i is the retrieval score of a given document and S_{max} and S_{min} are the maximum and minimum document scores by method i.

One of the main challenges in using the weighted fusion formula lies in determination of the optimum weights for each system (w_i). In order to optimize the fusion weights, WIDIT engages in a static tuning process, where various weight combinations are evaluated with the training data in a stepwise fashion.

4.5 Dynamic Tuning

Both topic and opinion reranking involve combination of multiple reranking factors as can be seen in the generalized reranking formula below:

$$RS = \alpha * NS_{orig} + \beta * \sum(w_i * NS_i) \quad (6)$$

In formula (6), NS_{orig} is the normalized original score, NS_{orig} is the normalized score of reranking factor i, w_i is the weight of reranking factor i, α is the weight of original score, and β is the weight of the overall reranking score.

To optimize the reranking formula, which involves determination of optimum reranking factor weights (w_i), we implemented *Dynamic Tuning* (Figure 2), which is a bio-feedback like mechanism that displays effects of tuning parameter changes in real time to guide human to find the local optimum.

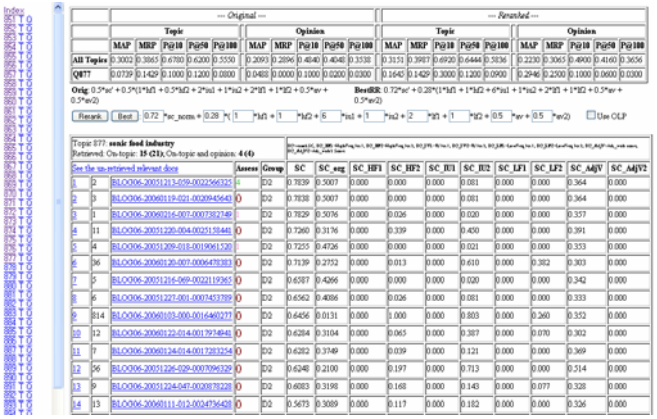

Figure 2: *WIDIT Dynamic Tuning Interface*

The key idea of dynamic tuning, which is to combine the human intelligence, especially pattern recognition ability, with the computational power of the machine, is implemented in a Web application that allows human to examine not only the immediate effect of his/her system tuning but also the possible explanation of the tuning effect in the form of data patterns. By engaging in iterative dynamic tuning process that successively fine-tune the reranking parameters based on the cognitive analysis of immediate system feedback, system performance can be improved without resorting to an exhaustive evaluation of parameter combinations, which can not only be prohibitively resource intensive with numerous parameters but also fail to produce the optimal outcome due to its linear approach to factor combination.

5. Experiment

Using the 2006 TREC Blog test collection, which consists of a Blog document set, 50 topics and associated relevance judgments, we generated a result set of 1000 blogs for each topic by first applying the topic reranking to initial retrieval results and then applying the opinion reranking. Fusion was applied to each of the reranking steps as well as to initial retrieval results.

5.1 Data

The Blog06 test collection includes a crawl of feeds (XML), associated permalinks (HTML, retrieval units), and homepages during Dec 2005 through early 2006. Among the blog document set 100,649 feeds (38GB), 2.8 million permalinks (75GB), and 325,000 homepages (20GB), only the permalinks were used in our experiment. 50 test topics, each consisting of title (phrase), description (sentence), and narrative (paragraph) fields, were constructed using queries from commercial blog search engines (e.g., BlogPulse and Technorati).

5.2 Relevance Judgments

TREC assessed a pool of unique results created from merging top 100 blogs from 27 submitted results and top 10 results from 30 submitted results. To be considered relevant, a blog had to be on topic and contain an explicit expression of opinion or sentiment about the topic, showing some personal attitude either for or against.

6. Results

The main performance evaluation metric for blog opinion retrieval task is mean average precision (MAP), which is the sum of precision at rank where relevant item is retrieved averaged over topics. Mean R-precision (MRP), which is the precision at rank same as the total number of relevant items averaged over topics, and precision at rank N (P@N) were also used to evaluate the system performances. The TREC official results of top 5 groups are displayed below.

Group	MAP	MRP	P@10
Indiana University	**0.2052**	**0.2881**	0.468
Univ. of Maryland	0.1887	0.2421	0.378
Univ. of Illinois at Chicago	0.1885	0.2771	**0.512**
Univ. of Amsterdam	0.1795	0.2771	0.464
Univ. of California, Santa Cruz	0.1549	0.2355	0.438

Table 1: *Official TREC blog opinion results of top 5 systems*

After the official submission, we conducted post-submission experiments that involved optimization of reranking and tuning modules using relevance data as well as overall system refinements. Among numerous system parameters at play, we examined the effects of following independent variables on retrieval performance using the post-submission results: query length, topic reranking, opinion reranking, dynamic tuning, and fusion.

6.1 Query Length Effect

It is well-know fact in information retrieval community that longer queries in general will produce better retrieval result. This was shown to hold true for blog opinion retrieval as well. Figure 3 shows consistently superior performances of longer queries in all phases of retrieval (i.e., initial retrieval, topic-reranking, topic reranking with dynamic tuning, opinion reranking, opinion reranking with dynamic tuning), and by both the topical and opinion performance evaluation. One exception occurs with baseline topic retrieval performance of the long query (title, description, narrative), which is worse than that of the short query. This may be due to introduction of noise in the long query, which is consistent with our past work that found some long queries to be harmful for finding specific targets due to introduction of noise [20]. When the same results are evaluated with opinion relevance (lower three line in Figure 3), however, the long query performs same as the short query. This suggests that the long query may contain description of opinions that helps finding opinion blogs while retrieving non-topical blogs at the same time. This anomaly is corrected by reranking strategy that uses combination of key evidences to boost the ranks of blogs likely to be relevant.

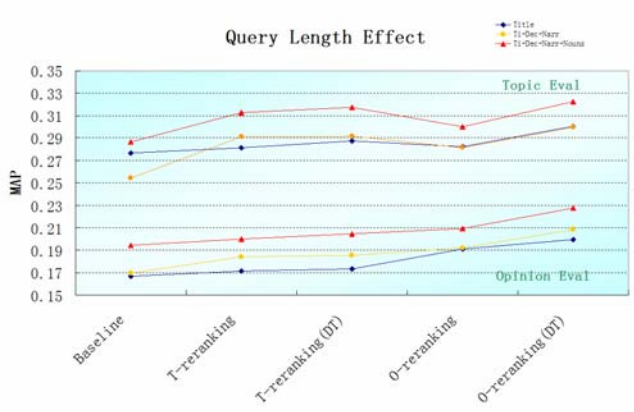

Figure 3: *Query Length Effect*

6.2 Topic Reranking Effect

The effect of topic reranking on initial retrieval is shown in Figure 4. The gain in topic retrieval performance by topic reranking is marginal for the short query (4%) but over 10% improvement for the long query. This is understandable since topic reranking factors capitalize on topical evidence, which the short queries have little of.

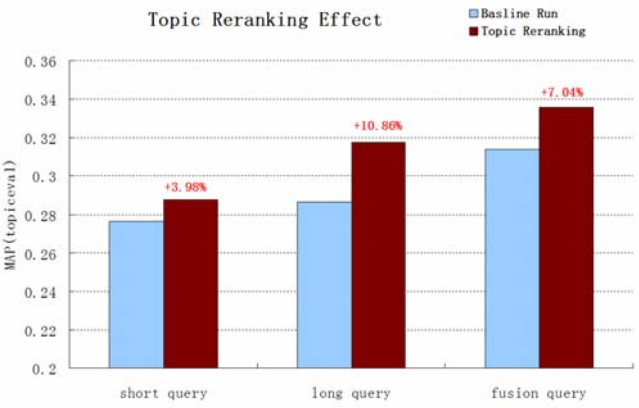

Figure 4: *Topic Reranking Effect*

6.3 Opinion Reranking Effect

Figure 5 displays the marked effects of opinion reranking. For the short query, opinion reranking improves the performance of topic reranked results by 15% (20% over baseline) and for the long query, 11% improvement (17% over baseline). It clearly demonstrates the effectiveness of WIDIT's opinion reranking approach.

6.4 Dynamic Tuning Effect

The effect of dynamic tuning is shown in Figure 6. Since the left bars show improvements over baseline that contain the reranking effect, the isolated effect of dynamic tuning turns out to be only marginal (4.5% for short query and 9% for long query). We suspect this is partially influenced by reranking effect that took the system performance towards the ceiling and partially by the mostly linear nature of tuned formulas that require more rule-based intervention to approach the optimum solution space.

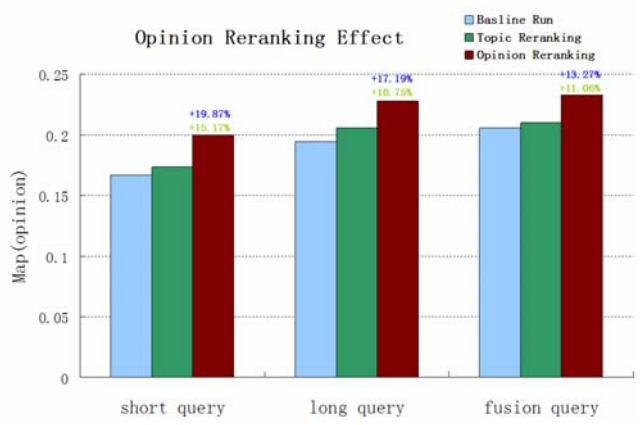

Figure 5: *Opinion Reranking Effect*

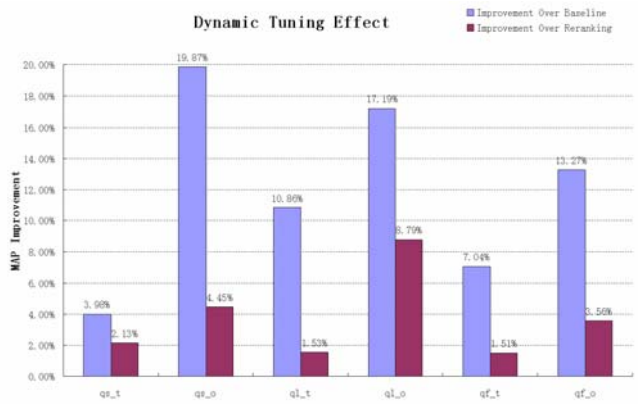

Figure 6: *Dynamic Tuning Effect*

6.5 Fusion Effect

As we have repeatedly found in previous research [20, 21, 22], the fusion approach is shown to be quite beneficial (Table 2). Fusion, which shows the best overall performance of all system combinations, improves performance by 20% over best baseline non-fusion result.

	QShort	QLong	Fusion
Baseline	.1666	.1943	.2057
Reranked - no Tuning - DTuning	 .1912 .1997	 .2093 .2277	 .2250 **.2230**

Table 2: *Opinion MAP of best baseline and fusion results*

7. Concluding Remarks

WIDIT's fusion approach of combining multiple sources of evidence and multiple methods worked well for TREC's blog opinion retrieval task. Topic and opinion reranking, as well as fusion all contributed to improving retrieval performance, and the compound effect of all three resulted in the best overall performance (dotted line in Figure 7). Although opinion retrieval posed non-trivial challenges, stepwise approach of initial retrieval, on topic retrieval optimization, and opinion identification proved to be an effective solution.

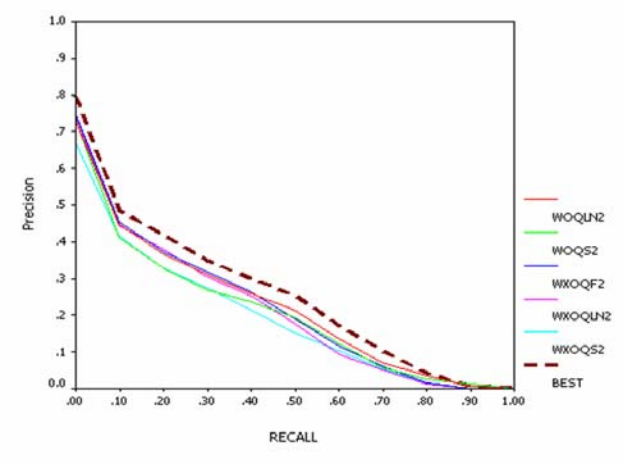

Figure 7: *Recall-Precision curve of WIDIT runs*

References

[1] Bartell, B. T., Cottrell, G. W., & Belew, R. K. (1994). Automatic combination of multiple ranked retrieval systems. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

[2] Buckley, C., Salton, G., & Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. *Proceeding of the 3rd Text Rerieval Conference (TREC-3)*, 1-19.

[3] Buckley, C., Singhal, A., & Mitra, M. (1997). Using query zoning and correlation within SMART: TREC 5. *Proceeding of the 5th Text REtrieval Conference (TREC-5)*, 105-118.

[4] Chklovski, T. (2006). Deriving quantitative overviews of free text assessments on the web. In IUI '06: *Proceedings of the 11th international conference on Intelligent User Interfaces*, New York, NY, USA, pp. 155–162. ACM Press.

[5] Efron, M. (2004). The liberal media and right-wing conspiracies: using cocitation information to estimate political orientation in web documents. *Proceedings of the thirteenth ACM international conference on Information and Knowledge Management*, 390–398.

[6] Fox, E. A., & Shaw, J. A. (1995). Combination of multiple searches. *Proceeding of the3rd Text Rerieval Conference (TREC-3)*, 105-108.

[7] Frakes, W. B., & Baeza-Yates, R. (Eds.). (1992). *Information retrieval: Data structures & algorithms*. Englewood Cliffs, NJ: Prentice Hall.

[8] Herring, S. C., I. Kouper, J. C. Paolillo, L. A. Scheidt, M. Tyworth, P. Welsch, E. Wright, and N. Yu (2005). Conversations in the blogosphere: An analysis "from the bottom up". *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*.

[9] Hu, M. and Liu, B. (2004). Mining and Summarizing Customer Reviews. In *KDD'04: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168–177.

[10] Lee, J. H. (1997). Analyses of multiple evidence combination. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 267-276.

[11] Liu, B., M. Hu, and J. Cheng (2005). Opinion observer: analyzing and comparing opinions on the web. *Proceedings of the 14th international conference on World Wide Web*, 342–351.

[12] Mishne, G. and M. de Rijke (2006). Deriving wishlists from blogs: Show us your blog, and we'll tell you what books to buy. *Proceedings of the 15th International World Wide Web Conference (WWW2006)*.

[13] Robertson, S. E. & Walker, S. (1994). Some simple approximations to the 2-Poisson model for probabilistic weighted retrieval. *Proceedings of the* 17th *ACM SIGIR Conference on Research and Development in Information Retrieval*, 232-241

[14] Savoy, J., & Picard, J. (1998). Report on the TREC-8 Experiment: Searching on the Web and in Distributed Collections. *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, 229-240.

[15] Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 21-29.

[16] Thompson. P. (1990). A combination of expert opinion approach to probabilistic information retrieval, part 1: The conceptual model. *Information Processing & Management*, 26(3), 371-382.

[17] Wiebe, J., T. Wilson, R. Bruce, M. Bell, and M. Martin (2004). Learning subjective language. Comput. Linguist. 30 (3), 277–308.

[18] Wilson, T., D. R. Pierce, and J. Wiebe (2003). Identifying opinionated sentences. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 33–34.

[19] Yang, K. (2002a). Combining Text-, Link-, and Classification-based Retrieval Methods to Enhance Information Discovery on the Web. (*Doctoral Dissertation*. University of North Carolina).

[20] Yang, K. (2002b). Combining Text- and Link-based Retrieval Methods for Web IR. *Proceedings of the 10th Text Rerieval Conference (TREC2001)*, 609-618.

[21] Yang, K., & Yu, N. (2005). WIDIT: Fusion-based Approach to Web Search Optimization. *Asian Information Retrieval Symposium 2005*.

[22] Yang, K., Yu, N., Wead, A., La Rowe, G., Li, Y. H., French, C., & Lee, Y (2005). WIDIT in TREC2004 Genomics, HARD, Robust, and Web tracks. *Proceedings of the 13th Text Retrieval Conference (TREC200*