WIDIT in TREC-2008 Blog Track: Leveraging multiple sources of opinion evidence

Kiduk Yang
School of Library and Information Science, Indiana University, Bloomington, Indiana 47405, U.S.A.
kiyang@indiana.edu

1. INTRODUCTION

Indiana University's WIDIT Lab[1] participated in the Blog track's opinion task and the polarity subtask, where we combined multiple opinion detection methods to leverage a variety of complementary evidences rather than trying to optimize the utilization of a single source of evidence.

To address the weakness of our past topical retrieval strategy, which generated mediocre baseline results with short queries (i.e., title only queries), we explored Web-based query expansion (WebX) methods to increase the initial retrieval performance for the short query. Our WebX methods consisted of the Google module, which harvests expansion terms from Google search results using various term selection methods, and Wikipedia module, which extracts noun phrases and related terms from Wikipedia articles and Wikipedia thesaurus. Web-expanded queries were combined using ad-hoc heuristics based on observation, trial and error, and some basic assumptions regarding quality of individual WebX methods.

For opinion detection, we leveraged the complementary opinion evidences of *Opinion Lexicon* (e.g., suck, cool), *Opinion Collocation* (e.g., I believe, to me), and *Opinion Morphology* (e.g., sooooo, metacool) in the form of semi-automatically constructed opinion lexicons. In creating the opinion lexicons, we extracted terms from external sources (e.g., IMDb movie reviews and plot summaries) as well as the blog collection. Opinion lexicons were utilized by opinion scoring modules to compute opinion scores of documents, and the combined opinion score in conjunction with the on-topic retrieval score was used to boost the ranks of opinionated documents. To optimize the fusion formula for combining opinion and polarity scores, WIDIT used the "Dynamic Tuning Interface", an interactive system optimization mechanism that displays the effects of tuning parameter changes in real time so as to guide its user towards the discovery of a system optimization state.

2. PROBLEM DEFINITION

Opinion finding task combines the problems of topical retrieval with opinion detection. The task of finding blogs that express opinion on a given target calls for an approach that can not only retrieve blogs about a target topic but also effectively measure the degree of opinion towards the topic. Opinion classification, which relies on the overall characteristic of a document to classify a document as opinionated without regard to the target of opinion, will produce false results when dealing with largely opinionated documents that contain no opinion towards the target topic (false positive) or mostly factual documents about a target with only brief expressions of opinion (false negative). Therefore, key strategies in opinion finding task are to optimize topical retrieval and to devise an effective opinion detection method that identifies opinion expressions associated with the target topic. For the task of topical retrieval, the strategy should be to optimize the differentiation of topically relevant documents from topically non-relevant ones rather than to optimize the ranking of the documents by the degree of topical relevance. In other words, the topical retrieval should be broad (i.e. recall-oriented) rather than specific (i.e. precision-oriented) with the provision that opinion detection will assist in increasing precision.

[1] Web Information Discovery Integrated Tool (WIDIT) Laboratory at the Indiana University School of Library and Information Science is a research lab that explores a fusion approach to information retrieval and knowledge discovery.

The opinion detection component of a targeted opinion detection system needs to first identify expressions of opinion and then to determine whether detected opinions are about a given target. An intuitive approach is to apply opinion detection at a subdocument level (e.g., paragraph, sentence) and look for the presence of target in proximity. At the subdocument level, opinion classification methods may be used in place of opinion detection since the overall characteristic of a document is strongly influenced by the presence of opinion evidence as a document gets shorter. Even so, the machine learning approach of opinion classification still has to contend with the scarcity of features in a short document. The association of opinion to a target by proximity is likely produce a high rate of true positives but may do so at the cost of false negatives. Opinion expressions that occur outside the proximity window, subdocument boundaries, or near the target that is described in a non-standard manner (e.g., synonyms, anaphors, spelling variations) will not be detected by the proximity method.

3. METHODOLOGY

Our approach to targeted opinion detection has two main components: a topical retrieval component and an opinion detection component. The topical retrieval process starts with an initial topic search to retrieve documents about a target topic, after which topical reranking is applied to optimize the ranking of the initial topic search result. The opinion detection component is comprised of multiple opinion scoring modules that leverage different sources of opinion evidence to maximize the coverage of opinion expressions.

Figure 1 displays the architecture of WIDIT blog opinion retrieval system. On the left side is the topical retrieval component and the right side shows the opinion detection component. The noise reduction, which was shown to be effective in 2007 (Yang, Yu & Zhang, 2007), is applied to exclude non-English blogs as well as non-content portions of each blog (e.g., navigation, advertisement). Web-based query expansion is applied to short queries to increase the initial topical retrieval performance, which is optimized by the *Topic Reranking Module* that utilizes topical clues not leveraged in the initial retrieval. Opinion scores computed by opinion detection modules are combined linearly to produce an overall opinion score of each document, which is used to rerank the topic-reranked results by the estimated degree of opinion. A set of opinion-reranked results with varying parameters (e.g., query expansion, opinion scoring) are combined to produce the final opinion result, to which polarity detection module is applied to generate the polarity result. To optimize topic reranking, opinion reranking, and polarity detection, all of which involve combining a large number of system parameters, we employed an interactive system optimization mechanism called *Dynamic Tuning* that harnesses human intelligence along with machine processing power in an iterative, bio-feedback like process to facilitate the discovery of local optimum solutions (Yang & Yu, 2005).

3.1. Web-based Query Expansion

Among the common query expansion strategies of pseudo-feedback, syntactic expansion by thesaurus (e.g., WordNet), and Web-based expansion, we chose the Web-based expansion to strengthen the short queries. Short queries of the blog opinion finding task tend to be one or two word descriptions of target entities and thus do not benefit much by syntactic expansion (e.g., synonym expansion). Pseudo-feedback, which relies on top ranked documents being relevant, can be problematic for short queries that produce poor initial retrieval results due to incomplete descriptions of entities. Web-based expansion, on the other hand, searches much larger external data sources of the Web, and has shown to be an effective query expansion strategy for difficult queries (Kwok, Grunfeld & Deng, 2005).

Our Web-based query expansion (QE) consists of the Wikipedia QE module, which extracts terms from Wikipedia articles and Wikipedia Thesaurus, and the Google QE module, which extends the PIRC approach that harvests expansion terms from Google search results (Kwok, Grunfeld & Deng, 2005).

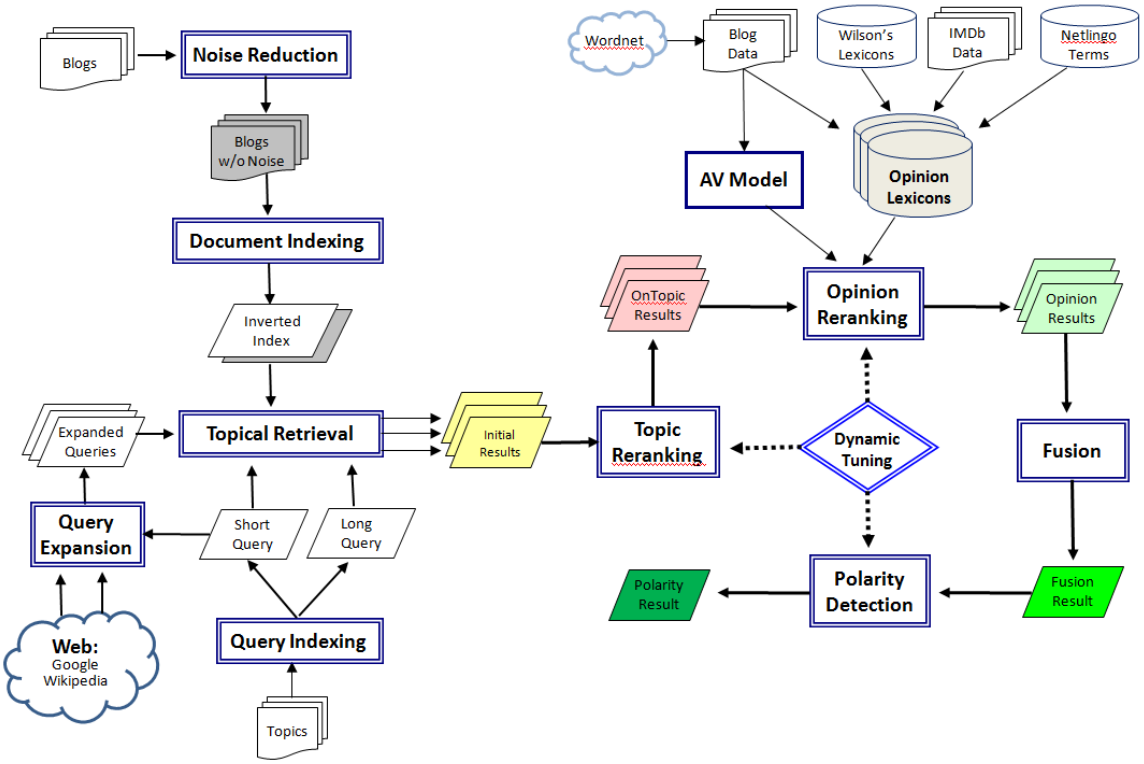

Figure 1. WIDIT Blog Opinion Retrieval System Architecture

3.1.1. Wikipedia QE (WQE) Module

WQE module generates two types of wiki-expanded queries. The first wiki-expansion starts by querying the Wikipedia search engine with the entire string of the short query. If the result is an encyclopedia article page, the title and top portion of the article up to the content listing is harvested for term selection. If the result is a full-text search result or disambiguation page that contains a list of potential matches, titles and snippets of the list items are harvested for further processing. The *k* most frequently occurring terms in the harvested text are added to the original query to create an expanded query.

The second type of wiki-expanded query consists of Wikipedia title and thesaurus terms. First, n-grams of decreasing length are extracted from the original query by a sliding window (e.g., "computer monitor price", "computer monitor", "monitor price", "computer", "monitor", "price") and checked against Wikipedia to identify phrases. The titles of Wikipedia pages ("e.g., "visual display unit", "price") retrieved by the longest n-grams (e.g., "computer monitor", "price") are then used to find synonyms and related terms from the Wikipedea Thesaurus[2]. Title terms from article pages are given the weight of 1, while title terms from search and disambiguation pages are given reduced weights (e.g., 0.8). The term weights of synonyms are even reduced further (1/2 of title term weights for synonyms, 1/3 for related terms). If the Wikipedia result is not an article, only the synonyms are added to reduce the adverse effect of incorrect expansion.

3.1.2. Google QE (GQE) Module

While WQE mines the manually constructed knowledge base of Wikipedia, GQE utilizes the rich information on the Web effectively searched by Google to identify related terms. Like WQE, GQE module

[2] http://wikipedia-lab.org:8080/WikipediaThesaurusV2/

generates multiple types of google-expanded queries by varying the source and weighting of expansion terms. The first step of GQE is to query Google with the short query and harvest the titles, snippets, and full-texts of the top *n* search results that are HTML pages. The first type of google-expanded query consists of top *k* most frequently occurring terms from titles and snippets. The second and third types of google-expanded queries are composed of top *k* weighted terms from the full-texts of search results, where the term weight is computed by a modified version of the local context analysis (LCA) formula (equation 2) using the original query and a combination of expanded queries.

The original LCA formula, shown in equation 1, selects expansion terms based on co-occurrence with query terms, $co(c,w_i)$, and their frequency in the whole collection, $idf(c)$, normalized over *n* (Xu & Croft, 2000). The *idf* component of LCA, which modifies the standard inverse document frequency with an upper bound, estimates the absolute importance of a term by its discriminating value, while the co-occurrence component estimates the relative term importance with respect to a given query. Since the collection frequency is unknown in the Web setting, we use term frequencies normalized by term distance to compensate for the lack of *idf*. The normalized frequency of term *c* in document *d*, is computed by first summing the word distances between occurrences of *c* and nearest query term w_i in *d* and taking the inverse of its log value. The normalized frequency of query term w_i in *d* is computed in a similar manner by taking the inverse log of the sum of minimum word distances between occurrences of query term w_i and *c* in *d*. The normalized term frequency modifies the weight of each term occurrence with co-occurrence distance in order to reward terms that occur closer to query terms.

$$\begin{aligned} co\_\deg ree(c,w_i) &= \log_{10}(co(c,w_i)+1)idf(c)/\log_{10}(n) \\ co(c,w_i) &= \sum_{d\ in\ S} tf(c,d)tf(w_i,d) \\ idf(c) &= \min(1.0, \log_{10}(N/N_c)/5.0) \end{aligned} \quad (1)$$

$$\begin{aligned} co\_\deg ree(c,w_i) &= \log_{10}(co(c,w_i)+1)/\log_{10}(n) \\ co(c,w_i) &= \sum_{d\ in\ S} tf_{norm}(c,d)tf_{norm}(w_i,d) \\ tf_{norm}(c,d) &= \sum_{c\ in\ d} \frac{1}{\log_5(\min dist(c,w_i)+1)} \\ tf_{norm}(w_i,d) &= \sum_{w_i\ in\ d} \frac{1}{\log_5(\min dist(w_i,c)+1)} \end{aligned} \quad (2)$$

3.2. OnTopic Retrieval Optimization

Optimizing the results of initial topic search is not only an efficient way to incorporate topical clues (e.g., phrases) not considered in initial retrieval (Yang et. al, 2007; Yang, Yu & Zhang, 2008), but also an effective way to supplement the recall-oriented initial retrieval by boosting the precision. The layered approach of first executing a recall-oriented search to produce retrieval results with high recall and then applying a post-retrieval reranking method to increases precision without degrading recall is a practical alternative to single-pass retrieval approaches that attempt to maximize the system performance by minimizing the recall-precision tradeoff (Buckland & Gey, 1994).

Our on-topic retrieval optimization method reranks the initial retrieval results based on a set of topic-related reranking factors, which consist of topical clues not used in initial ranking of documents. The topic reranking factors used in the study are: *Exact Match*, which is the frequency of exact query string occurrence in document, *Proximity Match*, which is the frequency of padded[3] query string occurrence in document, *Noun Phrase Match*, which is the frequency of query noun phrases occurrence in document, and

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

Non-Rel Match[4], which is the frequency of non-relevant nouns and noun phrase occurrence in documents. All the reranking factors are normalized by document length.

3.3. Opinion Detection

Our opinion detection approach, which is entirely lexicon-based to avoid the pitfalls of the machine learning problems, relies on a set of opinion lexicons that leverage various evidences of opinion. The key idea underlying our opinion detection strategy is to rely on a variety of complementary evidences rather than trying to optimize the utilization of a single source of opinion evidence. In keeping with the main research question of the study, we leveraged the complementary opinion evidences of *Opinion Lexicon* (e.g., suck, cool), *Opinion Collocation* (e.g., I believe, to me), and *Opinion Morphology* (e.g., sooooo, metacool) in the form of semi-automatically constructed opinion lexicons. Opinion lexicons are utilized by opinion scoring modules to compute opinion scores of documents, and the combined opinion score in conjunction with the on-topic score is used to boost the ranks of opinionated documents in a manner similar to the on-topic retrieval optimization.

Opinion scoring modules used in this study are *High Frequency module*, which identifies opinions based on the frequency of opinion terms (i.e., terms that occur frequently in opinionated documents), *Wilson's lexicon module*, which uses a collection-independent opinion lexicon based on Wilson's subjectivity terms, *Low Frequency module*, which makes use of uncommon/rare terms that express strong sentiments, *IU module*, which leverages n-grams with IU (I and you) anchor terms (e.g., I believe, You will love), *Opinion Acronym module*, which utilizes a small set of opinion acronyms (e.g., imho), and *Adjective-Verb Module*, which uses the density of potential subjective elements learned from training data to determine the subjectivity of documents. Each module except for the Adjective-Verb module computes three types of opinion scores for each lexicon used: a simple frequency-based score, a proximity score based on the frequency of lexicon terms that occur near the query string in a document, and a distance-based score computed as a sum of inverse word distances between lexicon terms and the query string. Equation 3 describes the generalized formula for opinion scoring

$$opSC(d) = \frac{\sum_{t \in L \cap D} f(t) \cdot s(t)}{len(d)} \qquad (3)$$

where L and D denote the term sets of a given lexicon and document d respectively, $len(d)$ is the number of tokens in d, $s(t)$ is the strength of term t as designated in the lexicon, and $f(t)$ is the frequency function that returns either the frequency of t in d (simple score), the frequency of t that co-occurs with the query string in d in a fixed-size window (proximity score), the sum of inverse word distances between occurrences of t and the nearest query string (distance score). The proximity score, which is a strict measure that ensures the opinion found is on target, is liable to miss opinion expressions located outside the proximity window as well as those near the target that is expressed differently from the query string. The simple score, therefore, can supplement the proximity score, especially when used in conjunction with the on-topic optimization. The distance score can be thought of as normalized frequency weighted by proximity similar to the normalized frequencies in the modified LCA formula (equation 2).

3.3.1. Polarity Detection

For polarity detection, positive and negative polarity scores are first computed by the opinion scoring modules using the score and polarity from lexicons. Equation below describes the generalized formula for computing opinion polarity scores:

[4] Non-rel Match is used to suppress the document rankings, while other reranking factors are used to boost the rankings.

$$opSC_{pol}(d)\frac{\sum_{t\in L_{pol}\cap D} f(t)\cdot s(t)}{len(d)} \qquad (4)$$

In equation 4, L_{pol} describes the lexicon term subset whose polarity is *pol* (positive or negative). The default term polarity from the lexicon is reversed if the term appears near a valence shifter (e.g., not, never, no, without, hardly, barely, scarcely, etc.) in *d*.

3.3.2. High Frequency Module

The basic idea behind the *High Frequency Module* (HFM) is to identify opinions based on common opinion terms. Since common opinion terms, which are words often used to express opinions, occur frequently in opinionated text and infrequently in non-opinionated text, a candidate HF lexicon can be constructed by identifying high frequency terms from the positive training data (i.e., opinionated documents) and excluding those that also have high frequency in the negative training data (i.e., objective documents). The resulting term set can then be manually reviewed to filter out spurious terms and to assign polarity and opinion strength.

In creating the HF lexicon, we used movie reviews and plot summaries to supplement the blog training data. The blog training data consist of 11,448 on-topic and opinionated (positive sample) and 8,301 on-topic but not opinionated (negative sample) blogs flagged in 2006 blog relevance judgments[5]. Since the quality of blog training data is degraded by document-level relevance judgments that are based on opinion detection rather than opinion classification, we decided to strengthen our lexicon base by adding terms extracted from a high quality sentiment-analysis data such as movie reviews[6] (positive sample) and a harvest of movie plot summaries (negative sample) from the Internet Movie Database[7] (IMDb). The final HF lexicon with 1,559 entries of opinion terms and associated polarity and strength were expanded with morphological variations such as inflections, superlatives, and alternate word forms (e.g., adjective to noun, adjective to adverb).

In addition to manually determined term weights, a second set of term weights that combine the manual weight with opinion probabilities of terms are computed using the training data. The combined weight formula, which is applied to all the lexicons, is shown in equation 5.

$$wt'(t) = wt(t)\frac{p(t\,|\,opD)}{p(t\,|\,nopD)} \qquad (5)$$

In equation 5, the probability of a term *t* occurring in an opinionated document, $p(t|opD)$, is estimated by the proportion of the documents with *t* in positive training data (i.e., opinionated documents), and the probability of a term *t* occurring in a non-opinionated document, $p(t|nopD)$, is estimated by the proportion of the documents with *t* in negative training data. To compute the combined weight, the manual weight, $wt(t)$, is multiplied by the probabilistic weight to adjust the human judgment of term importance with the training data evidence.

[5] Documents with more than 50,000 words were excluded.

[6] The movie review data was obtained from http://www.cs.cornell.edu/people/pabo/movie-review-data/

[7] http://www.imdb.com/Sections/Plots/

3.3.3. Wilson's Lexicon Module

To supplement the HF lexicon, which is collection-dependent and possibly domain-specific, we constructed a set of opinion lexicons from Wilson's subjectivity terms[8] (Wiebe et. al, 2004). *Wilson's Lexicon Module* (WLM) uses three collection-independent lexicons, which consists of 4,747 strong and 2,190 weak subjective terms extracted from Wilson's subjectivity term list, and 240 emphasis terms selected from Wilson's intensifiers. Both strong and weak subjective lexicons inherit the polarity and strength from Wilson's subjectivity terms, but the emphasis lexicon includes neither the strength nor polarity values since the strength and polarity of an emphasis term depend on the term it emphasizes (e.g., *absolutely* wonderful). In computing opinion scores (equation 3), emphasis terms are assigned the strength of 1, which is the minimum value for term strength. No polarity scores are generated with the emphasis lexicon.

3.3.4. Low Frequency Module

While HFM and WLM leverage the most common type of opinion evidence in standard opinion expressions, *Low Frequency Module* (LFM) looks to low frequency terms for opinion evidence. LFM is derived from the hypothesis that people become creative when expressing opinions and tend to use uncommon or rare term patterns (Wiebe et. al, 2004). These creative expressions, or Opinion Morphology (OM) terms as we call it, may be intentionally misspelled words (e.g., luv, hizzarious), compounded words (e.g., metacool, crazygood), repeat-character words (e.g., sooo, fantaaastic, grrreat), or some combination of the three (e.g., metacoool, superrrrv).

Since OM terms occur infrequently due to their creative and non-standard nature, we started the construction of the OM lexicon by identifying low frequency (e.g., $df < 100$) terms in the blog collection and excluding those that occur frequently in negative training data. Words with three of more repeated characters in the low frequency term sets were examined to detect repeat-character OM patterns, which were encapsulated in a compilation of regular expressions. Resulting regular expressions (OM regex) were refined iteratively as described below:

1. Apply OM regex to the low frequency term set.
2. Examine the results of step 1 to identify the false positives (non-opinion terms captured by OM regex) and false negatives (OM terms missed by OM regex).
3. Stop if there are no false positives and false negatives converge or fall below a threshold. Otherwise, go to step 2.

To round out the OM regex, regular expressions that simulate misspellings by vowel substitutions (e.g., luv) and regular expressions for capturing compound morphing were constructed from HF and Wilson terms, applied to the LF term set, and refined iteratively in the same manner as the repeat-character regex described above. LF terms not captured by OM regex but are nevertheless determined to be opinion terms form a basis for the OM lexicon. 352 entries in the final OM lexicon consist of opinion terms flagged during the OM regex construction process as well as those identified during the review of the LF term subset not matched by the final OM regex. The format of OM regex is consistent with other lexicons in that each entry is composed of a regular expression and associated polarity and strength. The OM regex contained 102 regular expressions of varying length.

3.3.5. IU Module

IU Module (IUM) is motivated by the observation that pronouns such as 'I' and 'you' appear very frequently in opinionated texts. IU collocations, which are n-grams with IU (I and you) anchor terms (e.g., I believe, you will love), mark adjacent statements as opinions (e.g., "*I believe* God exists", "God is dead *to*

[8] Wilson's subjectivity terms were obtained from http://www.cs.pitt.edu/mpqa/opinionfinderrelease/.

me"). In this regard, IU collocations provide yet another type of opinion evidence to complement the opinion lexicons of HFM/WLM and the opinion morphology of LFM.

For IU lexicon construction, we first extracted n-grams that begin or end with IU anchors (e.g., I, you, we, my, your, our, me, etc.) from the positive blog training data. Since IU collocations are not collection-dependent, movie reviews were also used to broaden and strengthen the IU n-gram harvest. Extracted n-grams were then manually filtered to create a lexicon of IU collocations (e.g., 'I believe', 'my assessment', 'good for you', etc.) with associated polarity and strength, after which the lexicon was expanded with additional IU collocations formed by coupling IU anchors with appropriate HF and Wilson terms (e.g., 'I' + verb, adjective + preposition + 'me'). As was done with the HF lexicon, morphological variations were applied to the final IU lexicon of 1037 entries (e.g., 'I think' to 'I think, thought, thinking'). In order to accommodate the various forms of an IU collocation (e.g., I believe, I cannot but believe, I have always believed, etc.), IUM applies the IU lexicon in a slightly different manner than other modules. First, documents are pre-processed to compress certain prepositions, conjunctions, and articles (e.g., of, with, and, or, a, the, etc.) as well as to convert IU texts with valence shifters to normalized forms (e.g., "I truly and seriously cannot" to "I-NOT") via regular expressions. Then IU collocations in the lexicon is "padded" in such a way that document texts with up to two words in-between IU collocation words will be matched by IUM (e.g., "I really truly think", "I am thinking").

3.3.6. Opinion Acronym Module

The Opinion Acronym lexicon used by *Opinion Acronym Module* (OAM) complements the IU lexicon with common IU collocation acronyms (e.g., imho). The OA lexicon consists of a manually filtered subset of Netlingo's chat acronyms and text message shorthand (http://www.netlingo.com/emailsh.cfm) in both acronym and expanded forms. Since opinion acronyms represent long phrases that serve as a clear indicator of opinion or sentiment, they were generally given higher opinion strength values than other lexicon entries. Like emphasis terms, no polarity was assigned to the 32 entries in the OA lexicon since they all turned out to be opinion indicators without orientation (e.g, jmtc for "just my two cents").

3.3.7. Adjective-Verb Module

While all preceding opinion modules are lexicon-based, the *Adjective-Verb Module* (AVM) attempts to learn the subjective language by utilizing adjectives and verbs with high distributional similarity to the seed adjective and verb set in the training data (Wiebe et. al, 2004). A small set of adjectives and verbs (e.g., good, bad, support, against, like, hate) expanded with synonyms and antonyms from lexical sources such as WordNet is used as a seed set to find a cluster of distributionally similar (i.e., high co-occurrence) adjectives and verbs in the positive training data. Based on the assumption that a document with a high concentration of subjective adjectives and verbs are likely to be opinionated, AVM uses the density of subjective adjectives and verbs, otherwise known as potentially subjective elements, to classify documents as opinionated or non-opinionated. A more detailed description of AVM can be found in (Yang et. al, 2007).

3.3.8. Opinion Reranking

After applying the opinion modules to top 500 documents of the topical (i.e., baseline) retrieval result, the weighted sum of the opinion scores are combined with the topical score to rerank the topical results. Equation 6 describes the reranking formula, where the min-max normalized original score of document d, $NS_{orig}(d)$, is combined with the weighted sum of opinion scores[9], $opS_i(d)$. α and β estimate the relative contributions of the original and combined opinion score in a way that enables the documents with high opinion score to float to the top without unduly influencing the existing document ranking. In the min-max

[9] Opinion scores are not min-max normalized since they are already document-length normalized.

normalization, (equation 7), $S_i(d)$ is the raw score of document d by component i, and $min\{S_i\}$ and $max\{S_i\}$ are the minimum and maximum scores by the component i.

$$RS(d) = \alpha * NS_{orig}(d) + \beta * \sum_{i=1}^{k} (w_i * opS_i(d)) \qquad (6)$$

$$NS_i(d) = \frac{S_i(d) - \min\{S_i\}}{\max\{S_i\} - \min\{S_i\}} \qquad (7)$$

3.4. Dynamic Tuning

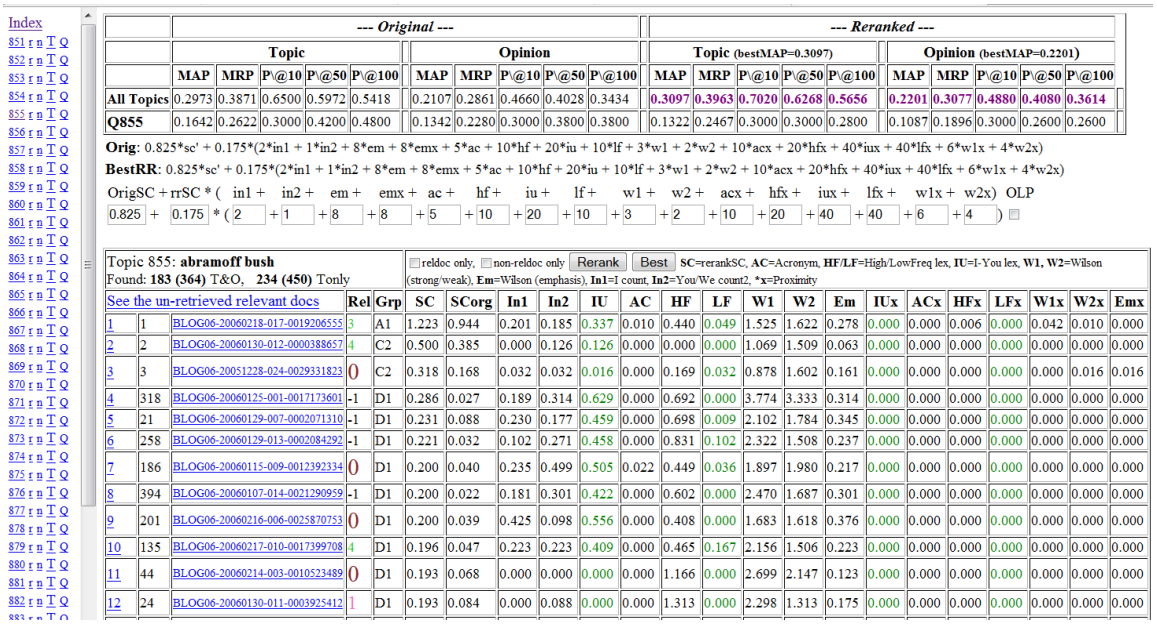

Figure 1. Dynamic Tuning Interface for Opinion Reranking Optimization

To facilitate the determination of fusion weights (w_i in equation 6) as well as original and reranking score weights (α and β in equation 6) in the reranking formula, we devised an interactive system tuning method called *Dynamic Tuning*. Dynamic Tuning is implemented in a Web-based interface that displays the effects of tuning parameter changes in real time so as to guide its user towards the discovery of a system optimization state in a biofeedback-like manner. To combine human intelligence, especially the pattern recognition ability, with the computational power of the machine, Dynamic Tuning allows the 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 that can be prohibitively resource intensive with a large number of parameters. Dynamic Tuning can also accommodate nonlinear combinations of factors that may arise from iterations of cognitive pattern analysis by incorporating them heuristically (e.g., decision rules) in conjunction with the weighted sum formula.

Figure 2 shows the dynamic tuning interface for optimizing the fusion formula that combines opinion reranking scores. The top portion of the interface displays in real time the effect of manually set fusion formula weights in terms of retrieval performance averaged over all topics as well as for the given topic. The bottom portion shows individual reranking factor scores for the ranked list of documents so that the human user may detect patterns that can be reflected in another cycle of tuning to beat the best performance (displayed in purple). Dynamic tuning can be thought of as a kind of human-driven genetic algorithm that

can facilitate searching of vast solution spaces by harnessing human's cognitive abilities to shortcut the purely algorithmic navigation towards the optimized solution. Since the processing power and speed of human falls far short of machine, however, the candidate solutions evaluated by dynamic tuning will normally span a much smaller area of the solution space than those covered by genetic algorithms. Consequently, dynamic tuning is likely to lead to only local optimum solutions. One way to compensate for this weakness is to integrate genetic algorithm with dynamic tuning so that human and machine can guide each other in the search of the optimal solution.

4. RESULTS

4.1. Performance Overview

Table 1 shows the performance of our opinion finding runs (IU-SLIS) in comparison with the best TREC official results (KLE, UIC_IR, fub) using short (i.e. title only) queries and standard baselines[10]. Our three runs in the table, which are identical in all aspects except for the baseline used, used manual lexicon weights with the reranking formula optimized with dynamic tuning. *Top3dt1mRd*, which used the combined result of the best three baselines, achieved the second best opinion finding performance. If we consider the performance improvement over baseline or runs using the baseline 4, however, our best run (*b4dt1mRd*) is ranked third.

Table 1: Opinion Finding results (Title only, Topics 1001-1050)

Group	Run	Baseline	Opinion MAP	Δ over Baseline
KLE	KLEDocOpinT	N/A	.4569	N/A
IU-SLIS	**top3dt1mRd**	2+3+4	**.4335**	**4.1%**
KLE	B4PsgOpinAZN	4	.4189	9.6%
UIC_IR	uicop2bl4r	4	.4067	6.4%
IU-SLIS	**b4dt1mRd**	4	**.4023**	**5.3%**
fub	FIUBL4DFR	4	.4006	4.8%

Table 2, which shows the performance of our polarity runs (IU-SLIS) along with top TREC official results (KLE, UIC_IR, fub) using short (i.e. title only) queries, clearly demonstrates the effectiveness of our polarity detection approach, especially in identifying negative opinions. We attribute this to our strategy of flagging valence shifters near opinion clues rather than attempting to determine the opinion polarity as a whole as must be done with machine learning approaches.

Table 2: Polarity Detection results (Title only, Topics 1001-1050)

Group	Run	Baseline	MixMAP (Δ)	Positive MAP (Δ)	Negative MAP (Δ)
IU-SLIS	**top3dt1mP5**	2+3+4	**.1677 (6.6%)**	**.1752 (2.3%)**	**.1601 (11.9%)**
KLE	KLEPolarity	N/A	.1662 (N/A)	.1828 (N/A)	.1496 (N/A)
IU-SLIS	**b4dt1mP5**	4	**.1572 (11.5%)**	**.1570 (2.5%)**	**.1574 (22.2%)**
KobeU	KobeU	4	.1448 (2.7%)	.1566 (2.2%)	.1329 (3.2%)
THUIR	THUpolTmfPNR	THUrelTwpmf	.1353 (7.2%)	.1289 (6.3%)	.1417 (8.0%)
UoGtr	UoGtr	4	.1348 (-4.4%)	.1394 (-9.0%)	.1301 (1.0%)
UWaterloo	UWaterloo	1	.1278 (0.7%)	.1430 (4.8%)	.1126 (-4.2%)

[10] KLE run with the best opinion finding MAP did not submit its baseline run.

4.2. Dynamic Tuning Effect

Figure 2 displays average performance improvements over baseline by opinion reranking (r2-r0), where green bars (s0R) indicate reranking without dynamic tuning and the red bars (s0R1) indicate reranking with dynamic tuning. The height of the bars represents the average performance difference between system pairs[11] that are identical in all aspects except for the parameter in question (e.g., baseline vs. opinion reranking with dynamic tuning). In addition to the beneficial effect of opinion reranking , which is clearly reflected in the upward direction of the bars across all performance measures (MAP, MRP, P@10), the chart also demonstrates the positive effect of dynamic tuning with the height differentials between runs with and without dynamic tuning. Markedly taller red bars indicate that dynamic tuning is quite effective in optimizing the reranking formula. For opinion reranking, the average gain in opinion MAP with dynamic tuning is almost 4 times that without dynamic tuning (20.03% vs. 5.89%).

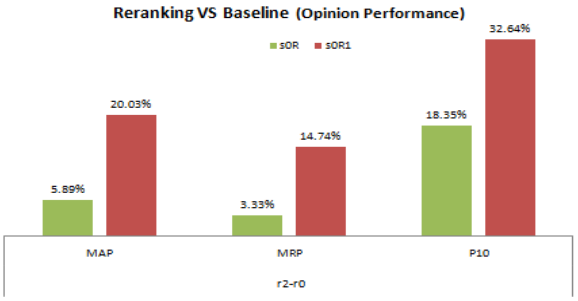

Figure 2: *Dynamic Tuning Effect*

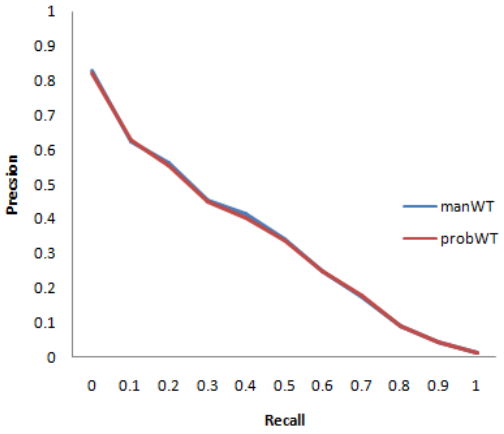

Figure 4: *Manual vs. Probabilistic Lexicon Weights*

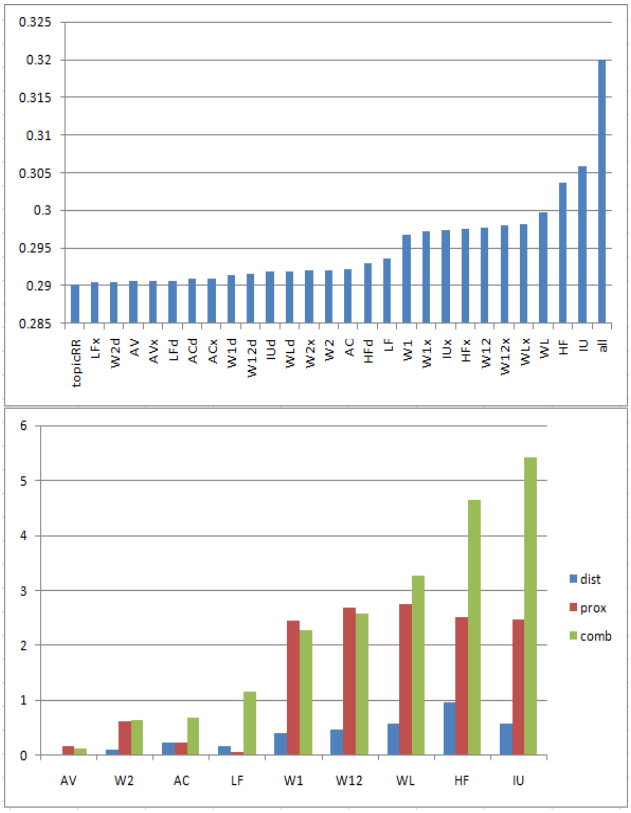

Figure 3: *Opinion Reranking Factors*

4.3. Opinion Reranking Factors

Charts in Figure 3 compare the effects of opinion evidences on opinion finding performance. In the chart on the left, which displays the individual performances of opinion reranking factors in an ascending order of effectiveness for the best short query run, "topicRR" bar shows the opinion MAP before opinion reranking, "all" shows the performance of the run that combines all opinion reranking factors with weights optimized via dynamic tuning, and the rest of the bars represent performances of single or combined opinion factors[12].

[11] System pairs consist of all our experimental runs including those that use our own baseline results.

[12] WLM's strong subjectivity terms are represented as 'W1', and weak subjectivity terms as 'W2'. 'W12' denotes both strong and weak subjectivity terms and 'WL' represents all Wilson's lexicon terms including emphasis terms.

The suffix 'x' of opinion factor names indicates frequency plus proximity scores, suffix 'd' indicates frequency plus distance scores, and neither 'x' or 'd' indicates combination of both. The chart on the right clusters reranking factors by type and plots the percent improvement in opinion MAP over topic reranked performance.

The charts show that all components of the opinion module contribute to opinion detection and the combined result outperforms the best single method by a significant margin (0.3199 vs. 0.3059: 46% increase). The best individual performance by the IU module suggests that the opinion collocation is a valuable source of evidence for opinion detection and thus should be leveraged along with the commonly used opinion lexicon evidence. The marginal performance gain by the AC module can be attributed to the small size of the acronym lexicon (32 entries). The poor performance of Adjective-Verb module could be affected by the manual intervention during AV lexicon expansion process. Manual filtering of lexicon terms that worked well with other lexicon-based approach may have inadvertently contaminated the construction of the AV model, which in theory should be learned from training data as whole regardless of individual term's semantic value.

It is easy to see from the bar cluster chart that the best source of opinion evidence is *Opinion Collocation*, followed by *Opinion Lexicon* (HF, WL) and *Opinion Morphology* (LF). The chart also shows that while proximity scoring is much better than distance scoring in general, combining both is quite beneficial. A notable exception is with LF, where distance scoring outperforms proximity scoring. This may be due to the usage pattern of opinion morphology that is less target-oriented than other opinion evidences. In other words, LF expressions may sometimes be used by themselves as exclamations of strong sentiments, whereas HF terms, for example, are more likely to be used in proximity of a target to denote targeted opinions (e.g., "I used Skype for the first time last night and it worked like a charm. *Grrrrreat*!" vs. "Skype is a *great* product.").

4.4. Manual vs. Probabilistic Lexicon Weights

Figure 4, which compares the recall-precision curve of the best short query run using the combined lexicon weights with a run that is identical in all respect except for the use of manual lexicon weights, shows that the outcome of using manual verses combined weights are nearly identical. This is probably due to the fact that combining probabilistic weights with manual weights by multiplication did not result in significant enough change in term weights as a whole to affect noticeable differences in outcome.

5. CONCLUDING REMARKS

Figure 5, which plots %MAP improvement over baseline of TREC participants' best runs[13], and figure 6, which plots %mixMAP improvement over baseline of TREC participants' best runs, give further evidence to the effectiveness of our approach of combining multiple complementary sources of evidence for opinion detection and demonstrate in an unequivocal fashion the effectiveness of our polarity detection approach that employs the detection of valence shifters in proximity of opinion terms with the polarity assignment at the lexicon level. The analysis of the results also reveals that Dynamic Tuning is an effective mechanism for optimizing the fusion of various opinion scores In addition to establishing an effective approach to targeted opinion detection that leverages multiple sources of evidence in an integrated fashion, we have also constructed opinion lexicons[14] for public use and presented a novel approach to system tuning that could prove useful for complex system optimization tasks.

[13] Red bars indicate our top 3 runs.

[14] http://elvis.slis.indiana.edu/lexlist.htm

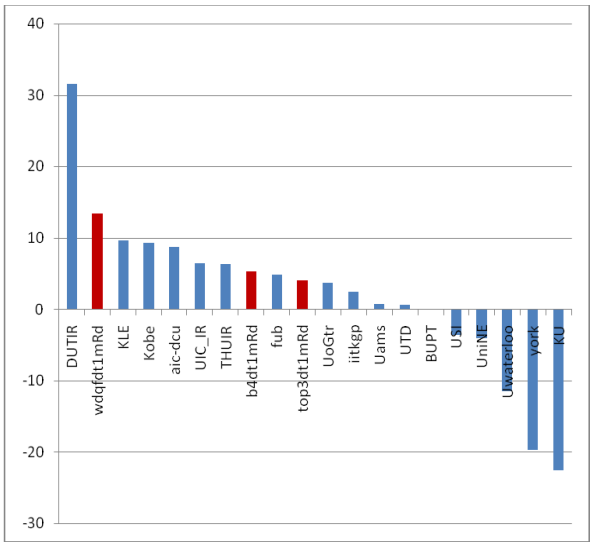

Figure 5: *%MAP Increase from Baseline, Best Opinion Finding Runs*

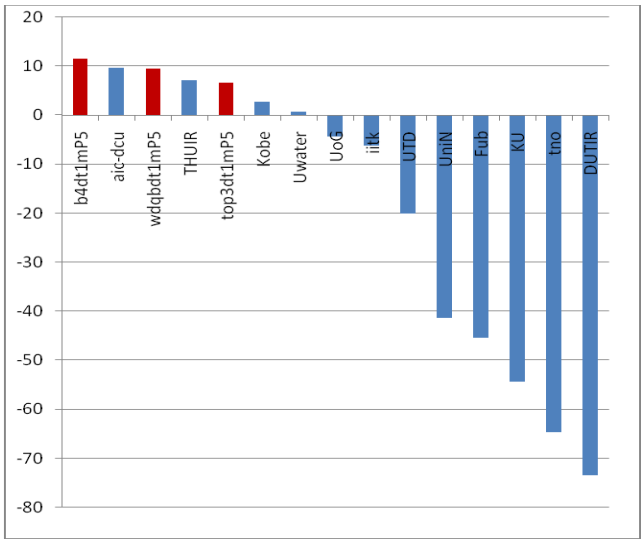

Figure 6: *%mixMAP Increase from Baseline, Best Polarity Runs*

REFERENCES

Buckland, M. & Gey, F. (1994). The Relationship between Recall and Precision, *Journal of the American Society for Information Science*, *45*(1), 12-19.

Kwok, K.L., Grunfeld, L., & Deng, P.(2005) Improving weak ad-hoc retrieval by Web assistance and data fusion. *Asian Information Retrieval Symposium 2005*.

Xu, J. & Croft, W.B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transaction on Information Systems, 18* (1), 79-112.

Yang, K., Yu, N., Valerio, A., & Zhang, H. (2007). WIDIT in TREC2006 Blog track. *Proceedings of the 15th Text Retrieval Conference* (TREC2006).

Yang, K., Yu, N., & Zhang, H. (2008). WIDIT in TREC2007 Blog track: Combining lexicon-based methods to detect opinionated Blogs. *Proceedings of the 16th Text Retrieval Conference* (TREC2006).

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

Yang, K., Yu, N., Valerio, A., Zhang, H., & Ke, W. (2007), Fusion approach to finding opinions in blogosphere. *Proceedings of the 1st International Conference on Weblog and Social Media*

Wiebe, J., Wilson,T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. *Computational Linguistics*, 30 (3), 277–308.