UIC at TREC 2010 Faceted Blog Distillation

Lifeng Jia and Clement Yu

Department of Computer Science

University of Illinois at Chicago

851 S Morgan St., Chicago, IL 60607, USA

{ljia2, cyu}@uic.edu

ABSTRACT

Our system consists of a concept-based retrieval subsystem which performs the baseline blog distillation, an opinion identification subsystem and an opinion-in-depth analysis subsystem which performs the faceted blog distillation task. In the baseline task, documents which are deemed relevant are retrieved by the retrieval system with respect to the query, without taking into consideration of any facet requirements. The feeds are ranked in descending order of the sum of the relevance scores of retrieved documents. In order to improve the recall of the retrieval subsystem, we recognize proper nouns or dictionary phrases without requiring matching all the words of the phrases. In the opinionated vs. factual and personal vs. official faceted tasks, the opinion identification subsystem is employed to recognize query-relevant opinions within the documents. Personal documents are more likely to be opinionated than official documents. In the in-depth vs. shallow faceted task, the depth of the opinion within a document is measured by the number of concepts which are related with the query the document contains.

1. INTRODUCTION

With the prevalence of Internet, more and more people express their opinions by writing online. Internet provides various textual resources covering a broad range of topics. Therefore, many researchers are interested in analyzing online text corpuses in depth, such as Blogosphere. Since 2006, TREC [8] has started a new track which provides Blogosphere collections to perform various text analysis techniques such as opinion retrieval [8, 9, 10, 17，15], polarity classification [9, 10, 15] and blog distillation [9, 10, 11]. Blog distillation was introduced in 2007 [9] and continued from 2008 to 2010 [10, 11]. In 2007 and 2008, blog distillation only focuses on the ad-hoc retrieval of feeds according to topical relevance. Since 2009, the facet blog distillation has been introduced. It addresses not only the topical relevance but also the quality of a given facet. These facets are paired into three groups:

1) Opinionated vs. Factual: Some bloggers may make opinionated comment on the topics of interest, while others report factual information. A user may be interested in blogs which show prevalence to opinionatedness. For this group, the values of facets are “opinionated' vs. “factual” blogs. [11]

2) Personal vs. Official: Companies are increasingly using blogging as an activity for PR purposes. However, a user may not wish to read such mostly marketing or commercial blogs, and prefer instead to keep to blogs that appear to be written by individuals without commercial influences. For this group, the values of facets are “personal” vs. “official” blogs. [11]

3) In-depth vs. Shallow: Users might be interested to follow bloggers whose posts express in-depth thoughts and analysis on the reported issues, preferring these over bloggers who simply provide quick bites on these topics, without taking the time to analyze the implications of the provided information. For this group, the values of facets are "in-depth" vs. "shallow" blogs (in terms of their treatment of the subject). [11]

2. BASELINE BLOG DISTILLATION TASK

To fulfill the baseline task which addresses only the topical relevance, an improved concept-based retrieval subsystem is utilized. The feeds are ranked in descending order of the weighted sum of topical relevance scores of retrieved documents belonging to them. The information retrieval subsystem has four components: concept identification, query expansion, concept based retrieval and document filters. We improved our last two components to enhance recall and precision.

2.1 Concept Identification

A concept in a query is a multi-word phrase or a single word that denotes an entity. Four types of concepts are defined: proper nouns, dictionary phrases, simple phrases and complex phrases. The proper nouns are names of people, place, event, organization etc, such as "*Hugo Chavez*". A dictionary phrase is a phrase that has an entry in a dictionary such as Wikipedia [13], but is not a proper noun, such as "*laser eye surgery*". A simple phrase is a 2-word phrase, which is grammatically valid but is not a dictionary entry or a proper noun, e.g. "*genealogical sources*". A complex phrase has 3 or more words but is neither a proper noun nor a dictionary phrase, such as "*United States future decline*". We developed an algorithm that combines several tools to identify the concepts in a query. We use Minipar [7], WordNet [14], and Wikipedia [13] and Google for proper noun and dictionary phrase identification. Collins Parser [2] is used to find the simple phrase and complex phrase. Web search engine (Google) is also used for identifying simple phrases within complex phrases. The details of the algorithm can be found in [16].

2.2 Query Expansion

Query expansion is another technique in the retrieval component. Two types of expansions are obtained: concept expansion and term expansion. In concept expansion, query concepts are recognized, disambiguated, if necessary and their synonyms are added. For example, for the query "*gun control DC*", there are many possible interpretations of "*DC*", according to Wikipedia. But, by using the query words "*gun control*", "*DC*" is disambiguated to "*Washington DC*", because "*gun control*" appears only in the Wikipedia entry of "*Washington DC*". As an example for concept expansion, consider the query "*alternative treatments for ADHD*". Proper noun "*ADHD*" has the synonym "*Attention Deficit Hyperactivity Discord*". Thus, the query becomes "*alternative treatments for ADHD*" OR "*alternative treatments for Attention Deficit Hyperactivity Discord*". Term expansion is carried out by the pseudo-feedback process in which terms in the vicinities of query terms in the top retrieved documents are extracted. We apply this technique to three different collections and take the union of the extracted terms. Specifically, the TREC documents and Web documents (via the use of Google) are employed. In addition, if a page in Wikipedia is found to represent a query concept, frequent words in that page are extracted. The union of terms extracted from these three sources is taken as the set of expanded query terms.

2.3 Concept-Based Information Retrieval

After concepts identification and query expansion, an original query will be augmented with a list of concepts and their synonyms (if exists) and additional words. In our information retrieval module, a query-document similarity consists of two parts: the concept similarity and the term similarity (*concept-sim*, *term-sim*). The *concept-sim* is computed based on the identified concepts in common between the query and the document. The *term-sim* is the usual term similarity between the document and the query using the Okapi formula [12]. Each query term that appears in the document contributes to the term similarity, irrespective of whether it occurs in a concept or not. The *concept-sim* has a higher priority than the *term-sim*, since we emphasize that the concept is more important than individual terms. Consider, for a given query, two documents D_1 and D_2 having similarities (x_1, y_1) and (x_2, y_2), respectively, where a x component represents concept similarity and a y component represents a term similarity.. D_1 will be ranked higher than D_2 if either (1) $x_1 > x_2$, or (2) $x_1 = x_2$ and $y_1 > y_2$. Note that if $x_i>0$, then the individual terms which contribute to *concept-sim* will ensure that $y_i>0$. The calculation of *concept-sim* is described in [4].

2.3.1 Relaxed Recognition of Proper Nouns and Dictionary Phrase

In this subsection, we present a new technique to retrieve documents which contain some, but not necessarily all component words of a proper noun query concept or a dictionary query concept. It is known that proper nouns and dictionary phrase appear frequently in user queries. For example, in TREC blog queries collected from 2006 to 2008, there are 114 proper noun queries and 14 dictionary phrase queries among 150 queries.

We first give an example to illustrate the idea. Consider the dictionary phrase *C* = "*Genome sequences*". Suppose there is a document containing the word *S* = "*Genome*", but without the word "*sequences*" next to it. This document may refer to a novel, instead of the hereditary information in molecular biology and genetics. To determine whether *S* in the document can be used to represent *C*, we proceed as follows:

(1) *S* is a prefix of *C*, if *C* represents a non-person proper noun or a dictionary phrase; if *C* is the name of a person, *S* is the last name;

(2) Both *C* and *S* are defined in Wikipedia;

(3) If *S* and *C* refer to the same entry in Wikipedia, then *S* can be used to represent *C* unambiguously; if *S* can refer to multiple entries in Wikipedia but the words in *C* – *S* can be used to uniquely identify the same entry as *C in Wikipedia*, then *S* is an ambiguous representation of *C*.

(4) If *S* is an ambiguous representation of *C*, a document *D* containing *S* must also contain at least one of the top two expanded terms of the documents containing *C* initially retrieved using the pseudo-feedback process or the terms in *C* – *S*; both *S* and one of these terms must be within a small window in *D*. If these conditions are satisfied, then *C* is assumed to be present in the document *D*.

For example, *C* = "*New York Philharmonic Orchestra*" and its prefix *S* = "*New York Philharmonic*" unambiguously refers to C according to Wikipedia. Therefore, "*New York Philharmonic*" can represent *C* without any constraints. However, in the "*Genome sequences*" example, Wikipedia has a lot of ambiguous entries about "*Genome*" but only the entry titled "*Genome*" has "*sequences*" in its content. Therefore "*Genome*" can refer to "*Genome sequences*" if at least one of the top two expanded terms

which are "*DNA*" and "*genetics*" or the query term "*sequences*" appears in close proximity with "*Genome*". As another example, suppose the user query is "*Hugo Chavez*". A document containing "*Chavez*" may or may not refer to the Venezuela President. It can be assumed to contain the query concept, if "*Chavez*" co-occurs in close proximity with at least one of the top two expanded terms "*Venezuela*" or "*President*" or the query term "*Chavez*". In the next subsection, we assign weights to such proper nouns, which are recognized in documents without exact matching all its component terms.

2.3.2 Relaxed Recognition of Proper Nouns

After a query concept is recognized in a document, it contributes a concept similarity to the document. If the document contains an ambiguous query concept, which can represent the query concept, the contribution will be reduced by a small value Δ, because there is a possibility that the representation is incorrect. We now determine the value of Δ such that the following condition is satisfied. Let D_1 be a document containing a subset $S_i = \{C_k\}$ of the set C of query concepts, D_2 be a document containing essentially the same set of query concepts $S_i' = \{C_k'\}$ where each C_k' is either an original query concept C_k in S_i or an ambiguous representation of C_k and at least one C_k' in S_i' is an ambiguous representation of C_k in S_i and D_3 be a document containing another subset S_j of C. If the concept similarity of D_1 is greater than that of D_3, then the concept similarity of D_2 is greater than that of D_3 while the concept similarity of D_1 is greater than that of D_2. D_1 should be ranked higher than D_2, because of the uncertainty of concept representation. D_2 should be ranked higher than D_3, because the ordering between D_1 and D_3 should be preserved by replacing D_1 by D_2, as D_2 has essentially the same set of query concepts as that of D_1. Supposed that the query "*Lance Armstrong*, *Alexander Vinokourov*" is submitted, the document D_1 containing both proper nouns will be ranked ahead of the document D_3 containing only "*Lance Armstrong*". The document D_2 containing "*Armstrong*" and "*Alexander Vinokourov*" and satisfying the relaxed form of "*Lance Armstrong*" should be ranked lower than D_1 but higher than D_3.

The following formula guarantees the above property. The justification is not given due to the lack of space. Given a query topic with a set of concepts, $C = \{C_1, C_2, \ldots, C_n\}$, let the concept weight due to C_i be W_i. For an ambiguous representation C_i', its concept weight is $W_i' = W_i - \Delta$, where Δ is computed by the formula below, where S_i and S_j are two subsets of C.

$$\Delta = \begin{cases} \dfrac{\min\limits_{m} \left\{ m = \left| \sum\limits_{S \in S_i} W_S - \sum\limits_{S \in S_j} W_S \right|, m > 0 \;\; S_i, S_j \in 2^C \right\}}{|C|} & , n \geq 2 \\ \dfrac{W_1}{2} & , n = 1 \end{cases}$$

2.4 Document Filters

Some techniques, such spam filters, are utilized to improve performance (especially precision) of concept-based document retrieval system. A spam component [15] is incorporated to filter out those spam documents, such as pornographic documents and non-English spam documents. Moreover, irrelevant documents where query terms only appear in some irrelevant portions of documents, such as advertisement or navigation bar, are also removed. [5] validated that the removal of irrelevant portions, such as advertisement, from a blog document can significantly improve the retrieval effectiveness within blogosphere. Advertisements usually have the following characteristics: 1) each advertisement is the contents of a leaf node in an HTML tree; 2) a number of advertisements are in common among

documents of the same feed, because a feed of documents is disseminated by the same content distributor. Thus, if the contents of a leaf node in a document are identical to that of a leaf node of another document in the same feed, then it is considered to be an advertisement. (In contrast, if a sentence in the main text of a document is identical to a sentence in a different document, but the sentence is not the entire contents of a leaf node, then the sentence is not recognized as an advertisement. In the main text, usually a paragraph or a sequence of paragraphs forms the contents of a leaf node.) Navigation bars within a document are adjacent hyperlinks and there are usually three or more such hyperlinks within the document. If query terms appear in advertisements or navigation bar portions of a document, they will not be used for retrieving the document.

2.5 Topical Relevance Ranking of Feeds

After documents are retrieved with respect to a query topic, a ranking of feeds will be generated. To demonstrate the rationale of ranking feeds according to the information of retrieved documents, some annotations are introduced first. Let q and f be a query topic and a feed respectively; D_q denotes the set of documents retrieved with respect to q and D_f is the documents of f; IR_D is the IR score of document D. For each feed, an aggregated score, S_f, is calculated as below and feeds are ranked according to descending order of this score.

$$S_f = \frac{|D_f \cap D_q|}{|D_f|} \times \sum_{D \in D_f \cap D_q} IR_D$$

3. FACETED BLOG DISTILLATION TASK

In faceted blog distillation task, six facets are identified and paired into three groups: opinionated vs. factual, personal vs. official and in-depth vs. shallow. In this section, we describe our opinion identification system. Then we propose the technique to measure the depth of an opinion within a document. Finally, we present the faceted feed ranking strategy.

3.1 Opinionated vs. Factual

A document is a relevant opinionated document with respect to a query topic, if it consists of at least one sentence, which is opinionated and is directed toward the query topic. We adopt the opinion analysis system from [15, 17]. The documents retrieved from the document retrieval system are classified into three categories: (1) factual documents; (2) opinionated documents but without topic-relevant opinion; and (3) opinionated documents with topic-relevant opinion. The opinion analysis system first utilizes a support vector machine (SVM-Light [3]) classifier to distinguish the documents of (2) and (3) from those of (1). Then, it employs a heuristic-based classifier to differentiate documents of (3) from those of (2). The opinion score of a document is the sum of the scores of its subjective relevant sentences provided by the SVM classifier and its similarity score. This yields an aggregate score. Then, opinionated documents are ranked in descending aggregate scores.

3.1.1 SVM-Based Opinion Classifier

A document is decomposed into sentences. Each sentence is classified by the SVM classifier to be either subjective (opinionated) or objective (factual). The document is opinionated if it has at least one opinionated sentence. In order to build the classifier, training data consisting of subjective data and

objective data are collected. Given a topic, topic relevant subjective documents are collected from review web sites such as Rateitall.com and Epinions.com. Additional documents are collected from the retrieved results of a search engine (Google) by submitting the query topic plus some "opinion indicator phrases" such as "*I like*" or "*I don't like*". The objective training documents are collected from Wikipedia. The dictionary entry pages of Wikipedia are considered to be high-quality objective data sources, as these pages describe things without opinion. The unigrams (individual words) and bigrams (two adjacent words form a bigram) extracted from the training data are the potential features to train the SVM classifier. We adopt the Pearson's Chi-square Test [1] to select the features. After the features are determined, each sentence from the training data is presented in a presence-of-feature vector, i.e. only the presence or absence of each feature is recorded in the vector, but not the number of occurrences of the feature. Then an opinion SVM classifier is established over that set of labeled vectors.

3.1.2 The *NEAR* Operator

After an opinionated sentence is identified by the opinion classifier, the opinion in the sentence may or may not be directed toward the query topic. The *NEAR* operator determines whether the opinionative sentence is pertinent to the query topic. Intuitively, an opinionative sentence has a good chance of being pertinent to the query topic, if the query terms appear in close proximity to the sentence. To be more specific, for each opinionative sentence, a text window of five sentences is set. The window consists of the opinionative sentence, two sentences preceding it and two sentences following it. In this paper, we give five conditions which determine whether an opinionated sentence is toward the query topic.

(1) The opinionative sentences which occur before the first appearance of a query concept are not considered as they are not pertinent to the query topic.

(2) If the query topic consists of one type of concepts which is either proper noun concepts or dictionary phrase concepts but not both types of concepts, then at least one concept must be matched and one term for each of the unmatched concepts must also be found in the text window. For example, for the query "*Drug Wars in Mexico*" with two proper noun concepts, "*Drug Wars*" and "*Mexico*", the opinionated sentence, "***Mexico*** *experiences a campaign of prohibition and foreign military aid being undertaken by the United States government, with the assistance of participating countries, intended to both define and reduce the illegal* ***drug*** *trade.* … ", is relevant although it can only match "*Mexico*" and only a content term, "*drug*" from unmatched "*drug wars*".

(3) If the query topic contains both proper noun concepts and dictionary phrase concepts, at least one proper noun concept must be matched and at least one term for each unmatched proper noun or dictionary phrase concept must be found in the text window. For example, for the query "*gun control DC*" with a proper noun concept, "*DC*" and a dictionary concept "*gun control*", the opinionated sentence about "…*Washington, D.C., has enacted a number of strict gun restriction laws…*" is relevant although it can only match "*D.C.*" and only a content term, "*gun*" from unmatched "*gun control*".

(4) If the query topic contains one type of concepts which is either proper noun or dictionary phrase concepts and additional content words, at least one proper noun or dictionary phrase concept and the content words must be matched and at least one term for each of unmatched concept must be found within the text window. For example, the query "*sciatica remedies*" with a dictionary phrase concept "*sciatica*" and one content word, "*remedies*", only the opinion concerning the

treatment aspect of "*sciatica*" is relevant to the query. Therefore, besides the dictionary phrase, "*remedies*" must be matched to guarantee the opinion is about the specific aspect of "*sciatica*".

(5) If the query topic without any proper noun or dictionary concepts, all query content words must matched. For example, a simple phrase, "*budget travel*", both content terms, "*budget*" and "*travel*", must be matched to guarantee the opinion is about the low-cost travel form, instead of general travel.

3.2 Personal vs. Official.

In our opinion, personal documents are more likely to be opinionated that official documents. Therefore, the same opinion identification system was employed in differentiating personal documents from official documents.

3.3 In-depth vs. Shallow.

A document which provides in-depth analysis about the query topic should not only be opinionated but also contains many concepts related with the query topics. A procedure is designed to obtain the set of concepts closely related to the query topic. For the convenience of introducing our technique, let us assume that any query topic can be represented by a set of proper noun or dictionary concepts, *C* and a set of content words, *T*. *RCC* denotes the set of related concepts candidates. The procedure defined below returns those k concepts which are most closely related to *q*.

Function Related_Concept_Recognition
Input: A query topic q = {C, T}; Parameter k.
Output: k concepts related to q

1. ***RCC =*** $\varnothing$ ***; // initialize***
2. ***for each concept c*** $\in$ ***C***
3. ***If c is defined in Wikipedia,***
4. ***RCC = RCC*** $\bigcup$ ***Wikipedia_Concept_Extractor(c).***
5. ***If RCC is not empty***
6. ***for each related concept candidate c*** $\in$ ***RCC***
7. PMI_c ***= Pointwise_Mutual_Information_by_Google(c, q).***
8. ***else // RCC is empty, q have no concepts defined in Wikipedia***
9. ***RCC = Google_within_Wikipedia(q)***
10. ***for each related concept candidate c*** $\in$ ***RCC***
11. PMI_c ***= Pointwise_Mutual_Information_by_Google(c, q).***
12. ***Rank related concept candidates according to descending order of PMI scores***
13. ***Return top k related concept candidates.***

The rationale of the function, "*Related_Concept_Recognition*", is to first locate a set of related concept candidates and then select those concepts which are closely related to the query topic, *q*, from those candidates by the Pointwise Mutual Information (PMI).

In line 1, *RCC* which stores all related concept candidates is initialized. In line 4, the function *Wikipedia_Concept_Extractor(c)* extracts the various types of concepts from the Wikipedia entry *c* and stores them in *RCC*. Proper noun and dictionary phrase concepts can heuristically be identified by those anchor texts which point to entries in Wikipedia. Moreover, simple and complex phrases are identified from the subtitles of Wikipedia entry of *c*, because subtitles normally convey related information concerning various aspect of the concept *c*.

From lines 5 to 7, each related concept candidate is assigned a weight equal to the PMI score which is estimated using documents retrieved by Google. The formula below is utilized to estimate the PMI score of a related concept candidate, *rc*, and the query topic *q*.

$$PMI(rc,q) = \log \frac{\frac{|GD(rc\ AND\ q)|}{|D|}}{\frac{|GD(rc)|}{|D|} \times \frac{|GD(q)|}{|D|}}, \quad \begin{array}{l} \textit{where } GD(x) \textit{ is the set of documents retrieved by Google w.r.t. } x \\ \textit{D is the whole set of documents indexed by Google} \end{array}$$

Lines 8-11 are intended for the situation when the query topic *q* has no concept defined in Wikipedia. For example, "*Budget Travel*" is a simple phrase query, but it is not defined in Wikipedia. The heuristic rule to locate the related concepts is to employ the parameterized Google search to retrieve documents from the site of Wikipedia and extract related concept candidates from the top 10 Wikipedia documents retrieved by Google. Then each related concept candidate is weighted by the PMI score of the query topic and it. For example, the top 10 Wikipedia documents retrieved by Google w.r.t. "*Budget Travel*" are shown and explained below. Among these 10 documents, 7 of them are related to the travel with budget or low cost and 3 of them are generally related with travel.

1) Backpacking(a form of low-cost, independent international travel);
2) Arthur Frommer (a travel writer whose writes a guide about budge travel);
3) Let's Go Travel Guides(the first travel guide series aimed at the student traveler);
4) CityPASS (A company that produces and sells booklets contain entrance tickets which is deeply discounted from the regular admission prices);
5) Hostel(provide information about budget oriented, social accommodation);
6) Low-cost carrier (airlines that generally has lower fares);
7) List of travel magazines;
8) Guide book(a book for tourists or travelers);
9) Rofl Potts(another travel writer);
10) Primera (an Icelandic Charter airline which provides budget travel operations);

After a set of related concepts is obtained, the extent of the depth of opinion within a blog document is measured by the sum of the normalized weights of related concepts which appear in it.

3.4 Facet Feed Ranking Strategy

A blog document is retrieved by the concept-based retrieval subsystem w.r.t. a query topic and then assigned a facet score w.r.t. the interested facet value. For the facet of "opinionated" (or "personal"), the relevant opinionated sentences within the blog document are identified by the opinion system and the facet score of "opinionated" (or "personal") is the sum of their SVM scores. For the facet of "factual" (or "official"), the facet score is the inverse of its facet score of "opinionated" (or "personal"). For the facet of "in-depth", the facet score is the sum of normalized weights of related concepts which appear in the document and the facet score for the "shallow" facet is the inverse of the "in-depth" score. After the facet score is calculated for a blog document, *d*, an aggregated score is obtained by linearly combining of its IR score and facet score as below.

$$AggregatedScore_d = a \cdot IRScore_d + (1-a) \cdot FacetScore_d$$

Let *q* and *f* be a query topic and a feed respectively; D_q denotes the set of documents retrieved with respect to *q* and D_f is the documents of *f*; AG_D is the aggregated score of document *D*. For each feed, a facet aggregated score, FS_f, is calculated as below and feeds are ranked according to descending order of

this score.

$$FS_f = \frac{|D_f \cap D_q|}{|D_f|} \times \sum_{D \in D_f \cap D_q} AG_D$$

4. EXPERIMENT EVALUATION

In this section, we evaluate the concept-based retrieval subsystem, opinion identification subsystem and opinion-in-depth system by Blogs08 Blogosphere collection and 63 of 100 queries released from 2009 to 2010. Only 39 of 50 queries from TREC 2009 contain at least one feed in both of two interested facets assigned to the queries. By now only 37 of 50 queries has been manually judged and only 24 of these manually-judged queries contain at least one feed in both two facets.

4.1 Baseline Blog Distillation

Table 1 shows the baseline performance of our concept-based retrieval subsystem. We utilize MAP, P@10, bPref and rPrec to measure the performance.

Table 1. The Performance of Baseline Blog Distillation

	MAP	P@10	bPref	rPrec
TREC 2009	0.2841	0.3974	0.3209	0.3459
TREC 2010	0.2036	0.3083	0.1947	0.2500

4.2 Facet Blog Distillation

In this section, we report the performance of our opinion identification system and opinion-in-depth system. We evaluate the MAP scores of rankings for six facets in Table 2.

Table 2. The Performance of Facet Blog Distillation

	Average	Opinion	Factual	Personal	Official	In-depth	Shallow
TREC 2009	0.1920	0.2175	0.1801	0.2190	0.1390	0.2678	0.1284
TREC 2010	0.1289	0.1080	0.1644	0.1088	0.1712	0.1125	0.1086

TREC coordinators also provide three topical baselines and suggest all participants employ their techniques on these baselines to measure the effectiveness of their techniques. In Table 3, we report the MAP scores of six faceted feed rankings on baselines with respect to 39 queries from TREC 2009. In table 4, the MAP scores of six faceted feed rankings on baselines are presented with respect to 24 queries from TREC 2010.

Table 3. Facet Blog Distillation on Baselines by TREC 2009 Queries

TREC 2009	Average	Opinion	Factual	Personal	Official	In-depth	Shallow
Baseline1	0.2193	0.2459	0.2183	0.2517	0.1603	0.2965	0.1433
Baseline2	0.1907	0.1970	0.1571	0.2221	0.1679	0.2725	0.1275
Baseline3	0.1698	0.1609	0.1442	0.1599	0.1754	0.2549	0.1233

Table 4. Facet Blog Distillation on Baselines by TREC 2010 Queries

TREC 2010	Average	Opinion	Factual	Personal	Official	In-depth	Shallow
Baseline1	0.1542	0.1650	0.1898	0.1192	0.2094	0.1259	0.1158
Baseline2	0.1435	0.1205	0.1729	0.1247	0.1966	0.1305	0.1160

Baseline3	0.1224	0.0933	0.1404	0.1040	0.1811	0.1034	0.1126

5. CONCULUSION

In this paper, we introduce the improved concept-based retrieval subsystem. A new technique is presented to improve the recall of the system by matching a concept within a document without requiring matching all content terms of that concept. An opinion identification subsystem and a technique to measure the depth of the opinions w.r.t. a query topic are demonstrated. The relevant opinions are identified by a SVM classifier and several heuristic rules. The depth of an opinion within a document is measured by the sum of weights of related concepts the document contains. The performances of the various systems are reported in detail.

6. REFERENCES

1. H. Chernoff and E. Lehmann. The use of maximum likelihood estimates in χ^2 tests for goodness-of-fit. The Annals of Mathematical Statistics. 1954.
2. M. Collins. Head-Driven Statistical Models for Natural Language Parsing. PhD Dissertation. 1999.
3. T. Joachims. Making large-scale SVM learning practical. Advances in Kernel Methods: Support Vector Learning. 1999.
4. S. Liu, F. Liu, C. Yu, and W. Meng. An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. In Proc. of SIGIR2004.
5. S. Nam, S. Na, Y. Lee and J Lee. DiffPost: Filtering Non-relevant Content Based on Content Difference between Two Consecutive Blog Posts. In Proc. of ECIR 2009, pp. 791-795

7. http://www.cs.ualberta.ca/~lindek/minipar.htm
8. I. Ounis, M. Rijke, C. Macdonald, G. Mishne, I. Soboroff. Overview of the TREC-2006 Blog Track. In TREC 2006.
9. I. Ounis, C. Macdonald and I. Soboroff. Overview of the TREC-2007 Blog Track. In TREC 2007.
10. I. Ounis, C. Macdonald, and I. Soboroff. Overview of the TREC-2008 Blog Track. In TREC 2008.
11. I. Ounis, C. Macdonald, and I. Soboroff. Overview of the TREC-2009 Blog Track. In TREC 2009
12. S. Robertson, S. Walker Okapi/Keenbow at TREC-8, 1999.
13. http://en.wikipedia.org
14. http://wordnet.princeton.edu/
15. W. Zhang, L. Jia, C. Yu and W. Meng. Improve the Effectiveness of the Opinion Retrieval and Opinion Polarity Classification. In Proc. of CIKM 2008, Napa Valley, CA. October 2008.
16. W. Zhang, S. Liu, C. Yu, C. Sun, F. Liu and W. Meng. Recognition and Classification of Noun Phrases in Queries for Effective Retrieval. In proceedings of the 16th CIKM. 2007.
17. W. Zhang, C. Yu and W. Meng. Opinion Retrieval from Blogs. In Proc. of CIKM 2007, Lisbon, Portugal. November 2007.