Identifying Facets in Query-Biased Sets of Blog Posts

Wouter de Winter
ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
The Netherlands
research@wouterdewinter.nl

Maarten de Rijke
ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
The Netherlands
mdr@science.uva.nl

Abstract

We investigate the identification of facets of query-biased sets of blog posts. Given a set of blog posts relevant to a topic, we compare several methods for identifying facets of the topic in this set. Building on a clustering of a set of blog posts, we compare several cluster labeling methods, and find that a method that makes use of blog and blog search specific features outperforms other methods. We also present efficiency-improving feature sets for clustering; our proposed method is fast enough to be deployed online.

1. Introduction

As the number of blogs grows and more people are involved in blogging, the need for effective blog retrieval increases. While search engines that focus solely on blogs already exist [2, 3, 4, 16], it is felt that new relevance models and new presentation modes are needed [11]. Queries submitted to blog search engines tend to be informational, with a strong focus on named entities and concepts [12]. With this comes the need to map the different facets of search results in a concise manner, especially because (like web searchers) blog searchers tend to inspect the first few search results only [12].

This, then, is the task we address in this paper: given a query-biased set of blog posts, identify important and distinct facets of the query present in the set. The scenario we envisage is that, alongside a ranked list of blog posts, a user of a blog search engine is presented with a list of facets in response to a query—allowing her to either go down the ranked list or to zoom in on one of the facets.

The identification of facets in a query-biased set of blog posts can be split in two: *clustering the blog posts* and *assigning labels to each cluster*. While the former is an essential prerequisite, our main focus in this paper is on the latter. Specifically, we take a standard clustering software package and experiment with different feature sets, guided mostly by efficiency concerns, so as to enable online deployment of our facet identification method. One of the challenges, both in the clustering and labeling phase, is the language used in blogs, which is problematic for labeling methods that are based exclusively on term frequencies [14, 20, 22]. We will introduce several labeling methods that go beyond mere term frequencies and introduce two new blog-specific ones.

The main contribution of this paper is an effective and efficient facet identification algorithm. The algorithm has been evaluated in a small-scale user study and it has been used online in an electoral search engine.

In the next section we describe related work. In Section 3 we detail our prototype, working on Dutch blog posts. In Section 4 the experimental setup is described and evaluation results are in Section 5. We conclude in Section 6.

2. Related Work

Access to blogs. With the launch of the blog retrieval track at TREC [13], research into information access to blogs received an important boost. Prior to that, Mishne [11] described blogs from an information access point of view, identifying the language used as an important and unique aspect. Mishne and de Rijke [12] analyzed query logs from blogdigger.com and found that most queries were informational in nature; users often look for the context of some named entity (what is said about certain persons, locations or organizations) and for high level topics, such as politics or culture. Like web searchers, they are only interested in the first few results and enter just a few queries per session.

Fujimura et al. [8] developed a multi-faceted search engine for blogs, Blogranger, offering Topic search, Blogger search and Blog search. Topic search supports informational queries, while the blogger and blog search facilities are suited for navigational queries. Their topic search is primarily based on named entities. After submitting a query, the system displays topics, organized by their named entity type.

Clustering and cluster labeling. Clustering can be done on an entire document collection or on a query-biased subset (query-specific clustering). Clustering the whole collection improves recall, but, with millions of documents and blogs available on the web, getting enough relevant documents is becoming easier and the focus shifts towards improving precision. Tombros et al. [19] use query-specific hierarchic clustering and show that query-specific clustering is more effective than more traditional static clustering. Another advantage of query-specific clustering is the lower computational demands.

A labeling method is a combination of generating candidate labels and choosing which candidates to use as cluster labels; labels can be multi-word units. Cutting et al. [5] use document titles as cluster names; Treeratpituk and Callan [20] use phrases instead of words, and Toda and Kataoka [18] use named entities to organize their labels. A common way to *find* labels is to take the most frequent words in a cluster (after removing stopwords). Cutting et al. [5] use this method in conjunction with titles of central documents in a cluster. This method tends to find 'collection stopwords:' stopwords

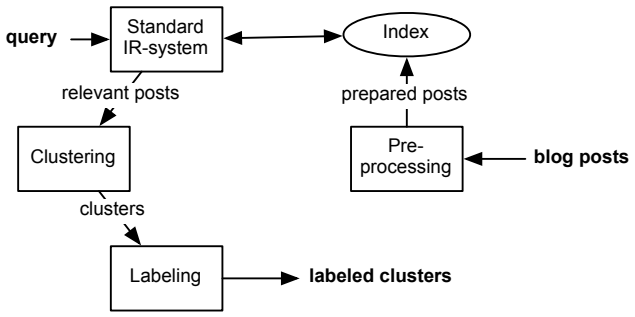

Fig. 1: *System architecture of a protoype facet identification method system*

for the specific collection that carry little semantic value. Another method is to take the most predictive or salient words from a cluster. Rather obscure and infrequent words, along with misspellings are likely to be chosen as cluster labels. To overcome the problems of these two methods, Popescul and Ungar [14] combine them to find labels in a hierarchic clustering system, reporting improvements over existing methods.

Balog et al. [2] use the log-likelihood statistical test to discover overused words in sets of blog posts sharing a similar emotion indicated by the bloggers themselves; Dunning [6] describes the theoretical background of this method.

3. A Facet Identification Method

To assess how well a baseline approach to facet identification works on (query-biased sets of) blog posts, and how well blog-specific extensions perform, we built a prototype system. After detailing it below, we zoom in two aspects: the features used in clustering, and labeling methods.

3.1 Architecture

Our facet identification system (visualized in Figure 1) has four main parts: a *pre-processor*, a (standard) *document retrieval system*, a (standard) *clustering package*, and a *labeler*. Here is a brief description of the main components:

a. Pre-processor. Standard text pre-processing is performed, which includes stemming, and stopword removal; phrases and named entities (NE) are recognized and indexed.

b. Standard information retrieval system. We use Lucene [7] for its performance and adaptability.

c. Clusterer. Organizes the posts into clusters. We use the CLUTO toolkit [9], with the default $k-1$ repeated bisections clustering method (with $k = 10$).

d. Labeler. The labeler examines the posts in a cluster and assigns a meaningful label to it.

3.2 Features for clustering

Instead of comparing different clustering methods (as in [9, 23, 24]), we focus on the features to be used for clustering blog posts. We compare three feature sets: stemmed words, named entities and overused terms.

Baseline. Here, we use a *bag-of-words* approach, where the frequencies of all terms are taken as the features for clustering. To reduce the words to their stem (and conflate e.g., singluar and plural forms) we use the Snowball stemmer for the Dutch language [15]. Stemming has the additional advantage of feature reduction. To further reduce the number of features, rare terms are pruned and stopwords are removed. The feature set that remains after stemming and stopword removal is our *baseline feature set*. Instead of working with raw term frequencies, we use binary features: the presence of a term in a post is only counted once. This is done for all methods, including the baseline. Early experiments show that this reduces sensitivity for noise. From initial experiments it seemed quite sensitive for noise. The noise causes the clustering algorithm to cluster the posts on other aspects than topic alone, e.g., by language, or by author.

Named entities. Blog searchers are mainly interested in blog posts about named entities [12]; searches for persons are the most popular. In the first of two alternatives to the baseline feature set we use the named entities as features for clustering. Posts dealing with the same persons, locations or other named entities are clustered together. When a user searches for a person, she instantly sees to what other persons or locations this person is related. We use the named entity recognizer developed in [17], with its Dutch language file.

Overused terms. Another way to eliminate noise is to restrict the features to terms which are important for the topic. To this end we use a method similar to one used by Balog et al. [2]. We compare two sets of posts, the posts relevant for the topic (from the Standard-IR system) and the complete set of posts. To find "overused words," we calculate the log-likelihood score for each word. The n-highest scoring terms are chosen as the features for clustering. This method reduces the influence of noise on clustering. Terms which are not important for the topic do not affect the clustering process. Another advantage of this method is the dramatic reduction in feature size, thus improving efficiency; see below.

3.3 Labeling methods

With clusters as input, the labeler produces a ranked list of candidate labels with a confidence score for each cluster. The highest scoring label (per cluster) is chosen as the cluster label. Then duplicate detection is performed. If one of the cluster labels is also a label for another cluster, only the label with the highest confidence is retained. If a label is removed, a new candidate label is added. This continues until no duplicates are left.

We consider six methods for creating labels: *term frequency*, *log-likelihood*, *extended log-likelihood*, *phrases*, *named entities*, and *mutual information*.

Term frequency. After removing stopwords, the most frequent term of a cluster becomes its label [5].

Log-likelihood. We use the log-likelihood method to detect overused terms in a cluster (comparing the posts in a cluster with the complete set of posts) and select the overused terms in the cluster as labels.

Extended log-likelihood. By to the log-likelihood method a term may be overused with respect to the complete set of posts but not compared to posts in other clusters (in the query biased set of posts): such a term is important for the topic but has no importance within the topic. To address this we calculate the log-likelihood score over the posts in the cluster, the relevant posts and the complete set of posts. A term is overused if it is occurs more than expected compared

to the relevant posts and the complete set of posts.

Phrases. Some concepts (e.g., people's names) are better represented as n-grams than as single words. To create useful and meaningful n-grams we use a chunker [17], which we limit to noun phrases. The phrase labeler extracts the noun phrases similar to the phrase clustering method. These phrases are the candidate labels. The extended log-likelihood test is then applied to determine the most overused phrase and this phrase is chosen as the cluster label.

Named entities. We use the named entity recognizer developed from [17], producing named entities as candidate labels. The extended log-likelihood test is used to choose the most overused named entity from the candidates.

Mutual information. Another way to overcome noise is to use external knowledge. In the methods discussed so far, labels are selected based on statistics of the blog corpus. Especially when a topic has few posts, these statistics are easily affected by noise and strange labels are chosen. A common, data-driven way to perform a 'sanity-check' is to use web search engines to calculate the Pointwise Mutual Information (PMI) [10, 21]. The extended log-likelihood method is used to select the 10 most overused terms from a cluster. The PMI-IR measure is then calculated for these terms and the highest scoring term is selected as the cluster label.

4. Experimental set-up

We assess the impact of the two stages (*clustering* and *labeling*) on the overall facet identification task. Our focus is on precision (What fraction of the 10 facets identified are good facets?), and we want to see to which extent duplicate results and incompleteness or vagueness of the labels is an issue.

Below, we describe our dataset, the task being we aim to evaluate, the topics used, and the measures aimed at capturing different aspects of the facet identification task.

Dataset. For our experiments we use a collection of blogs made available by the Dutch website `web-log.nl`. Covering January 2006, there are 34,122 active blogs in the set, with 367,129 posts in total, and 10.75 posts per blog, on average. The number of unique terms is 860,840, of which 540,287 occur only once.

The task. The system's task is to return, for a given input topic, and a query-biased set of blog posts, a list of 10 facets identified in the query-biased subset. This is also the task that we assessed, and for which topics and metrics were developed; see below. We did not evaluate the retrieval and clustering stages of our facet identification methods—because both are only intermediate stages in our setup and because cluster evaluation is hard.

Topics. Thirty topics were developed; thirty seems a reasonable balance between the resources we had available for assessment efforts and the number usually required for observing statistically significant differences. Based on [12], we created informational topics only, 18 named entities (person, location, organization) and 12 concepts (6 general, 6 specific).

Assessments. For assessment purposes we recruited four assessors, each with considerable search engine and blog search experience. For a given label proposed by the system the assessors were asked to judge whether it captures an facet of the topic at hand. For each topic-list-of-proposed-facets generated by of the methods, the assessor was asked to determine whether the cluster is a facet of the topic. She could also indicate that a facet is a duplicate. The label can be incomplete, for instance when referring only to the surname of a person instead of the complete name. When a label is unreadable, not in Dutch or the terms used are not known to the assessor, she judges the facet as *unclear*.

Every assessment was performed by two users. The Cohen's Kappa score for these assessors was .60 (p=.000). This is regarded as a good inter-rater reliability [1].

Label/Clustering method	Baseline	Named entities	Overused words
Baseline	.29	.29	.36
Log-likelihood	.45	.49	.47
Extended log-likelihood	.55	.56	.56
Named entities	.48	.49	.47
Phrases	.56	.57	.55
Mutual information	.60	.60	.60

Table 1: *Average precision scores for all feature set/labeling method combinations.*

Metrics. In line with our main focus, our chief evaluation metric is precision@10. We let each combination of clustering and labeling method generate a list of ten clusters. The assessor then examines these clusters. The clusters are represented by the cluster-labels. Below, we report on a small-scale efficicency experiment where we measure the executing a single test topic in milli-seconds.

Significance testing. We use the Wilcoxon Matched-Pairs Signed-Ranks (two-tailed) test to test for significance.

5. Results

We present results on effectiveness, and on efficiency.

Efficiency. The feature sets defined in Section 3 differ in size, and, hence, will have different impacts on the efficiency of the cluster algorithm, and on the efficiency of the facet identification process, where clustering is computationally the most expensive stage. We performed our tests on an Intel Pentium M processor (1.73GHz) with 1.50GB internal memory, and observed the following response times per topic (averaged over our 30 test topics, which were run 10 times): 1,366 ms for the baseline feature set (320,000 features), 512 ms for the named entities feature set (153,312 features), and 144 ms for the overused feature set (100 features). The size of the latter feature set is an arbitrary choice: 100 worked well in practice. In this case, the response time of the whole system is about 600 ms; other parts of the system are not optimized.

Effectiveness. We compared three features sets for clustering and six labeling methods. The feature sets are: baseline (stemmed terms), named entities, and overused terms. The labeling methods are: baseline (term frequency), log-likelihood, extended log-likelihood, phrases, named entities and mutual information. An overview of the precision scores for all 18 combinations is given in Table 1. We see substantial differences between the different labeling methods, but, somewhat surprisingly, there appears to be very little difference between the feature sets used: the use of the strongly reduced (and much more efficient) "overused words" feature set does not come at the price of a reduction in precision

	Prec	Dupl	Inc	Uncl
Baseline	.37	.00	.09	.15
Log-likelihood	.47	.01	.08	.10
Extended log-likelihood	.56	.03	.10	.07
Named entities	.48	.03	.03	.11
Phrases	.55	.05	.01	.06
Mutual information	.60	.04	.13	.06

Table 2: *Average scores for all labeling methods and all metrics, using the "overused words" feature set for clustering. (Prec: Precision; Dupl: Duplicate; Inc: Incomplete; Uncl: Unclear)*

	Baseline	LL	ELL	NE	Ph
Log-likelihood (LL)	.0033*				
Ext. log-likelihood (ELL)	.0001*	.0029*			
Named entities (NE)	.0065*	.5677	.0215		
Phrases (Ph)	.0001*	.0068*	.7897	.0051*	
Mutual information	.0000*	.0012*	.1683	.0049*	.1245

Table 3: *Significance tests for all pairs of labeling methods, using the "overused words" clustering feature set. * denotes a significant difference in per-topic precision scores ($p < .01$).*

(compared to the other two feature sets). In one case (using the term frequency labeling baseline), the reduced feature set actually leads to an improvement in precision.

Given these observations, we base additional reports on the "overused words" feature set only. From the labeling methods, the baseline method performs worst (Table 2); it produces many unclear labels. These labels are often numbers or single letters and due to the noisy nature of blogs these terms often occur in blog posts. This method also has a tendency to select 'collection stopwords' as labels. The Log-likelihood and Extended log-likelihood methods improve on the baseline. Many of incompleteness phenomena are overcome by the Named entities and Phrases labeling methods. The use of multi-word labels also introduces an inherent risk of generating duplicates. The Mutual information labeling method achieves the highest precision scores; it also has the largest fraction of incomplete labels.

In Table 3 we provide a pairwise comparison of the labeling methods. All methods perform significantly better than the baseline. The Mutual information method also outperforms both the Log-likelihood and the Named entities method.

6. Conclusion

We addressed the task of identifying facets in a query-biased set of blog posts, and decomposed it into a clustering and label generation stage. For the former we compared three feature sets, and found that the blog-specific "overused words" feature set was as effective as feature sets that are larger by several orders of magnitude. On top of this, an extended log-likelihood based method, complemented with sanity-checking based on pointwise mutual information proved the most effective label generation method.

The reduction of the size of the clustering feature set improved the efficiency of the clustering stage. The complete process—including facet identification—can be performed in a few hundred milliseconds, showing that online facet identification in query-biased set of blog posts is feasible.

Acknowledgments

We thank Ilse Media B.V. for providing us with the dataset and are very grateful to our assessors, Welmoed Fokkema and Leon de Winter. Maarten de Rijke was supported by the Netherlands Organization for Scientific Research (NWO) under project numbers 017.001.190, 220-80-001, 264-70-050, 354-20-005, 600.065.120, 612-13-001, 612.000.106, 612.066.302, 612.069.006, 640.001.501, 640.002.501, and by the E.U. IST programme of the 6th FP for RTD under project MultiMATCH contract IST-033104.

7. References

[1] R. Bakeman and J. M. Gottman. *Observing interaction: An introduction to sequential analysis.* Cambridge UP, 1986.

[2] K. Balog, G. Mishne, and M. de Rijke. Why are they excited? In *Proc. EACL 2006*, 2006.

[3] Blogdigger. Search engine for RSS and blogs, January 2006. URL: `http://blogdigger.com/`.

[4] Blogpulse. Automated trend discovery system for blogs, January 2006. URL: `http://blogpulse.com/`.

[5] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *Proc. SIGIR*, pages 318–329, 1992.

[6] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 1993.

[7] A. S. Foundation. Lucene homepage. url: `http://lucene.apache.org/java/docs/index.html`, September 2006.

[8] K. Fujimura, H. Toda, T. Inoue, N. Hiroshima, R. Kataoka, and M. Sugizaki. Blogranger - a multi-faceted blog search engine. In *Proc. 3rd Annual WWE*, 2006.

[9] G. Karypis. CLUTO clustering toolkit manual, release 2.1.1. Technical report, Univ. Minnesota, Dept. Comp. Sci., 2003.

[10] G. Mishne. Experiments with mood classification in blog posts. In *Style2005 – 1st Workshop on Stylistic Analysis of Text for Information Access, at SIGIR 2005*, 2005.

[11] G. Mishne. Information access challenges in the blogspace. In *IIIA-2006 – Intern. Works. Intell. Inform. Access*, 2006.

[12] G. Mishne and M. de Rijke. A study of blog search. In *Proc. ECIR 2006*, pages 289–301, 2006.

[13] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the TREC-2006 Blog Track. In *TREC 2006 Working Notes*, pages 15–27, November 2006.

[14] A. Popescul and L. H. Ungar. Automatic labeling of document clusters. In *Proc. KDD'00*, 2000.

[15] M. Porter. Snowball stemmer homepage, 2006. URL: `http://www.snowball.tartarus.org/`.

[16] Technorati. Blog tracking service, 2006. URL: `http://technorati.com/`.

[17] E. F. Tjong Kim Sang. Noun phrase detection by repeated chunking. In *CoNLL-99*, 1999.

[18] H. Toda and R. Kataoka. A clustering method for news articles retrieval system. In *WWW '05: Special interest tracks and posters*, pages 988–989, 2005.

[19] A. Tombros, R. Villa, and C. V. Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Inform. Proc. & Management*, 38(4):559–582, July 2002.

[20] P. Treeratpituk and J. Callan. Automatically labeling hierarchical clusters. In *Proc. of the Sixth National Conference on Digital Government Research*, pages 161–176, 2006.

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

[22] H. J. Zeng, Q. C. He, Z. Chen, W. Y. Ma, and J. Ma. Learning to cluster web search results. In *Proc. SIGIR'04*, pages 210–217, 2004.

[23] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical Report #01–40, Dept. Comp. Sci, Univ. Minnesota, 2001.

[24] Y. Zhao and G. Karypis. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, 2005.