Ranking Opinionated Blog Posts using OpinionFinder

Ben He, Craig Macdonald, Iadh Ounis
Department of Computing Science
University of Glasgow, Scotland, UK
{ben,craigm,ounis}@dcs.gla.ac.uk

ABSTRACT

The aim of an opinion finding system is not just to retrieve relevant documents, but to also retrieve documents that express an opinion towards the query target entity. In this work, we propose a way to use and integrate an opinion-identification toolkit, OpinionFinder, into the retrieval process of an Information Retrieval (IR) system, such that opinionated, relevant documents are retrieved in response to a query. In our experiments, we vary the number of top-ranked documents that must be parsed in response to a query, and investigate the effect on opinion retrieval performance and required parsing time. We find that opinion finding retrieval performance is improved by integrating OpinionFinder into the retrieval system, and that retrieval performance grows as more posts are parsed by OpinionFinder. However, the benefit eventually tails off at a deep rank, suggesting that an optimal setting for the system has been achieved.

Categories and Subject Descriptors: H.3.3 [Information Storage & Retrieval]: Information Search & Retrieval

General Terms: Performance, Experimentation

Keywords: Opinion finding, Blogs

1. INTRODUCTION

The advent of the opinion finding task in the TREC Blog track [2, 4] has increased research interest in the retrieval of opinionated blog posts. Two opinion identification approaches appear to be effective: firstly approaches based on the presence of opinionated terms in the retrieved documents, and secondly, looking for more complex subjective sentences structures, as identified using machine learning techniques. We propose an approach that belongs to the latter. It is based on OpinionFinder (OF), which is a freely available toolkit for identifying subjective sentences in text [8]. Two Naive Bayes classifiers are applied that distinguish between subjective and objective sentences using a variety of lexical and contextual features. The classifiers have been trained using subjective and objective sentences, which have been automatically generated from a corpus of un-annotated data by rule-based classifiers [7]. In this paper, we propose applying OF for retrieving opinionated blog posts, and show how simple evidence from OF can be utilised for effective opinion finding performance. Moreover, as OF is a computationally expensive technique to apply, we vary the number of retrieved blog posts for which we apply it, to measure its effect on opinion finding retrieval performance.

SIGIR'08, July 20–24, 2008, Singapore.
ACM 978-1-60558-164-4/08/07.

2. INTEGRATING OPINIONFINDER WITH RANKED RETRIEVAL

For each retrieved blog post, up to a given rank, we parse the document, including comments, using OpinionFinder (OF). We then examine the accuracy-oriented classifier of OF on each sentence of a given post. In particular, we consider the confidence value of OF's classifier of the identified subjective sentences, and the proportion of the identified subjective sentences in each considered blog post. We assign an opinion score to each blog post, and combine the opinion score with the relevance score given by the weighting model used, to produce a final relevance score for ranking the retrieved blog posts.

Intuitively, the subjectivity of a blog post increases with the percentage of the subjective sentences in the blog post. Therefore, we define the opinion score $Score(d, OF)$ of a document d produced by OF as follows:

$$Score(d, OF) = sumdiff \cdot \frac{\#subj}{\#sent} \quad (1)$$

where $\#subj$ and $\#sent$ are the number of subjective sentences and the number of sentences in the document, respectively. *sumdiff* is the sum of the *diff* value of each subjective sentence in the document, showing the confidence level of subjectivity estimated by OF. Such an opinion score is then combined with the relevance score $score(d, Q)$ given by the weighting model used, to produce the final relevance score.

Inspired by the weight-normalised Compliment Naive Bayes in [6], our combination method maps each opinion score to a probability $P(opn|d, OF)$ of being opinionated as follows:

$$P(opn|d, OF) = \frac{Score(d, OF)}{\sum_{d \in R(d,OF)} Score(d, OF)} \quad (2)$$

where $R(d, OF)$ is the set of all documents on which OF has been applied. Since a high $P(opn|d, OF)$ indicates a high degree of opinion expressed in the document, we would like to have a combined score that is an increasing function of $P(opn|d, OF)$. Therefore, such a probability $P(opn|d, OF)$ is combined with the initial relevance score as follows:

$$Score_{com}(d, Q) = \frac{-k}{log_2 P(opn|d, OF)} + Score(d, Q) \quad (3)$$

where k is a free parameter.

3. EXPERIMENTS AND ANALYSIS

We base our experiments on the Blog06 collection created for the TREC Blog track [3], which is currently the only available Blog test collection with relevance assessments.

We use the Terrier IR platform for both indexing and retrieval [5]. We index only the blog posts and their associated comments, as this is the retrieval unit of the TREC opinion finding task. Each term is stemmed using Porter's stemmer, and standard English stopwords are removed. Moreover, we index each field (content, title and anchor text of incoming hyperlinks) separately, and use the PL2F field-based document weighting model. PL2F is a combination of the Divergence from Randomness (DFR) PL document weighting model and Normalisation 2F for weighting fields [1].

We use the 100 title-only topics from the TREC 2006 & 2007 opinion finding tasks, numbered from 851 to 950. We use the 50 topics from the opinion finding task in 2006 for training, to set the parameter k of Equation (3) and the parameters of PL2F [1]. In particular we use $k = 100$. For evaluation on the 50 topics from TREC 2007, we use opinion finding MAP, where the retrieved posts must not only be relevant to the query but also express an opinion about the target [2, 4].

The purpose of our experiments is to evaluate our proposed opinion blog post retrieval method using OF. In particular, we examine to which extent the effectiveness of our proposed approach is affected by the parsing depth, i.e. the number of the top returned documents parsed by OF. On top of the retrieval baseline, i.e. the PL2F weighting model, we apply our proposed method using OF described in Section 2, with different parsing depths of the retrieved documents. The parsing depths tested increase from 10 to 1200 with an interval of 10. For example, if parsing depth is 20, only the top 20 returned documents for each query are parsed using OF.

However, using OF is very computationally expensive. Indeed, the parsing of 150 documents takes approximately 1 hour of CPU time of a Pentium III 1GHz node (NB: We improved the efficiency of OF, and these improvements are now part of version 1.5). Hence to perform our experiments, many such machines were applied to parse all of the retrieved blog posts.

Figure 1 plots the parsing depth and its corresponding retrieval performance. Parsing depth zero is equivalent to our retrieval baseline using PL2F only. From Figure 1, we observe a strong correlation between the parsing depth and the resulting MAP retrieval performance - i.e. MAP increases with OF's parsing depth. The linear correlation between MAP and the parsing depth is $\rho = 0.9997$, which indicates an almost perfect correlation. It is also encouraging to see that our proposed approach outperforms the baseline with all the different parsing depths applied (from baseline 0.2817 to 0.3301 MAP). In contrast, for P@10, applying OF to just the top 60 ranked documents results in an increase of 22% over the baseline, while applying OF to more documents has no further effect on P@10. Applying OF to only the top 10 ranked documents results in a statistically significant increase in P@10 (Wilcoxon Signed Rank Test, $p \leq 0.05$), while applying to the top 60 documents results in a significant increase in MAP.

Figure 1 also shows the parsing time in CPU hours on 1 Pentium III 1GHz processor for various parsing depths - the time taken is strictly linear. However, while a marked improvement in performance is observed for parsing only 200-400 posts, as the depth of OF increases the growth of MAP decreases, especially when compared to the constant growth in parsing time. Indeed, after about 1000 posts there is very little improvement in MAP for parsing more posts, while parsing time continues to grow linearly. Note that

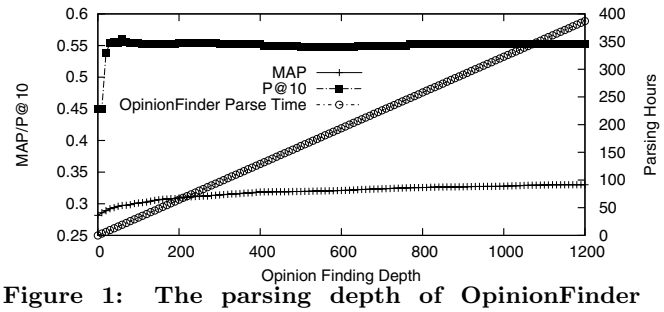

Figure 1: The parsing depth of OpinionFinder against the resulting MAP and parsing time.

with over 3.2 million posts in the Blogs06 test collection, it remains unfeasible to pre-parse all blog posts at indexing time - indeed, 21,000 CPU hours would be required!

4. CONCLUSIONS

In this paper, we have proposed an effective method for blog post opinion retrieval, which uses OpinionFinder (OF), a NLP-based toolkit for identifying subjectivity in text. The OF tool, and our proposed method to integrate it to a ranked retrieval task are both shown to be very effective in our experiments on the TREC Blog track opinion finding task. We find it promising that OF performs well on this setting, which is a much larger document corpus than used in previous evaluations of OF, although efficiency issues remain. Moreover, while parsing more posts with OF results in higher retrieval performance, there is a tail-off in MAP growth above rank 1000, suggesting that this is probably the optimal setting for the system. However, this may be partly due to the limited effect that a document ranked at 1200 is likely to have on an evaluation that terminates at rank 1000 (as in normal TREC evaluation). Finally, if only early precision is important, then applying OF to less top-ranked documents (e.g. 60) balances effectiveness and efficiency. The results shown in this poster are comparable with the 2nd best run title-only in the TREC 2007 Blog track opinion finding task [2], and given a stronger baseline retrieval system (e.g. using external query expansion [2]), could potentially improve further. In the future, we plan to investigate ways to improve the efficiency of OF without reducing its effectiveness in opinion finding.

5. REFERENCES

[1] D. Hannah, C. Macdonald, B. He, J. Peng, and I. Ounis. University of Glasgow at TREC 2007: Experiments in Blog and Enterprise Tracks with Terrier. In *Proceedings of TREC 2007*.

[2] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2007 Blog Track. In *Proceedings of TREC 2007*.

[3] C. Macdonald, and I. Ounis. The TREC Blog06 Collection : Creating and Analysing a Blog Test Collection *DCS Technical Report TR-2006-224*. University of Glasgow. 2006.

[4] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the TREC 2006 Blog Track. In *Proceedings of TREC 2006*.

[5] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of OSIR 2006*.

[6] J. Rennie, L. Shih, J. Teevan, and D. Karger. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings of ICML 2003*.

[7] E. Riloff and J. Wiebe. Learning Extraction Patterns for Subjective Expressions. In *Proceedings of EMNLP 2003*.

[8] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. OpinionFinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demos*, 2005.