DCU at the TREC 2008 Blog Track

Adam Bermingham
CLARITY: Centre for Sensor Web Technologies
and Centre for Digital Video Processing
Dublin City University
Dublin, Ireland
abermingham@computing.dcu.ie

Alan Smeaton
CLARITY: Centre for Sensor Web Technologies
and Centre for Digital Video Processing
Dublin City University
Dublin, Ireland
asmeaton@computing.dcu.ie

Jennifer Foster
National Centre for Language Technology
Dublin City University
Dublin, Ireland
jfoster@computing.dcu.ie

Deirdre Hogan
National Centre for Language Technology
Dublin City University
Dublin, Ireland
dhogan@computing.dcu.ie

ABSTRACT

In this paper we describe our system, experiments and results from our participation in the Blog Track at TREC 2008. Dublin City University participated in the adhoc retrieval, opinion finding and polarised opinion finding tasks. For opinion finding, we used a fusion of approaches based on lexicon features, surface features and syntactic features. Our experiments evaluated the relative usefulness of each of the feature sets and achieved a significant improvement on the baseline.

1. INTRODUCTION

This was Dublin City University's first year participating in the Blog Track at TREC and this marks our return to TREC participation after a layoff of several years[1]. We aimed to evaluate the effectiveness of combining three different approaches to opinion finding. Machine learning has been used successfully in the field of Sentiment Analysis [3], [9] and each of our approaches uses machine learning techniques to generate sentiment scores for relevant blog entries.

Our system consists of:

- A **Lexicon** module which evaluates the sentiment orientation of a blog entry by aggregating the sentiment scores for its consituent words in a sentiment lexicon, SentiWordNet.
- A **Surface** module which scores documents based on textual features which are obtained without any parsing or syntactic understanding of the sentence structure.
- A **Syntactic** module which scores documents based on features derived from part-of-speech tagging and parsing the text.

We fuse the scores from each of the modules using weighted combsum late fusion [11]. We have also submitted runs which do not weight the various sources, as a comparison.

Three baselines were used in our experiments: a topic only retrieval run, a topic and description run and "baseline4" as distributed by TREC. We chose baseline4 as time did not permit us to perform experiments on more than one distributed baseline and it gave a higher MAP on the previous two years' queries than the other four distributed baselines.

The rest of this paper is organized as follows: The system is described in Section 2. The runs submitted and results are discussed in Section 3. Conclusions are then presented in Section 4.

2. SYSTEM DESIGN

As is common in previous years' opinion retrieval submissions [8], [5], we favoured a design based on a two-stage system. The first stage concerns retrieving and ranking topic relevant blog entries. In the second stage the results are re-ranked according to opinion scores (or polarised opinion scores for the polarity task).

2.1 Relevant Blog Retrieval

The open source retrieval platform Terrier [7] was used to index the collection and retrieve query-relevant blog entries. The retrieval model used was the Okapi BM25 model. We submitted two relevance baselines: DCUCDVPtbl, our title only run, and DCUCDVPtdbl, our title and description run.

During system development we experimented with various different query expansion models available on the Terrier platform before settling on the Divergence From Randomness Bo1 (Bose-Einstein 1) model. We tuned the query expansion parameters based on retrieval MAP scores for the topics from the Blog Track 2006 and 2007. For the title-only run we found that setting the number of documents to look for query expansion to 2, and the maximum number of query terms to 7, gave optimum results. For the topic and description baseline, we did not find that adding additional terms to the query improved performance. We did however observe that improved performance could be obtained for this baseline by using the query expansion model to re-

[1]We have been participants in TREC from the early 90s to early 00s

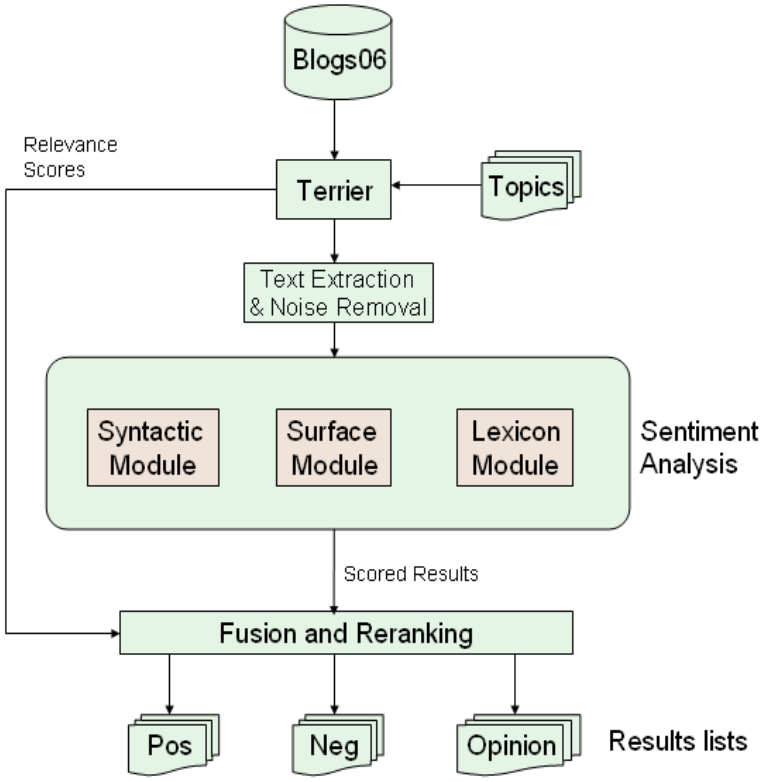

Figure 1: DCU's system architecture

weight the terms in the query. For the title and description baseline we therefore looked in the top 7 documents but set the number of terms to add to 0.

2.2 Feature Extraction

For each topic, a ranked list of 1000 documents is returned to the feature extraction stage. Firstly, the HTML documents referred to by the results lists are parsed using HTMLParser [1] to extract the natural language from the blog entry. The documents are separated into text sections corresponding to blocks of text delimited by HTML elements which are deemed to break the flow of the text by the parser. Certain sections are then judged noisy if they have a high link-to-text ratio (e.g. advertisements, link lists) or if they have a high non-alphabetic character to alphabetic character ratio (e.g. code citation, date). These sections are removed before feature extraction.

From the text we then extract three types of features which are described in the following sections.

2.2.1 *Surface Features*

We chose to use a set of surface features as we wanted to evaluate features which may be derived easily and which offer information not necessarily contained in a standard bag-of-words model. The vector of surface features is comprised of four different types of feature:

- A number of document measurements. Looking at logical lengths in the document such as section length or word length, we derived a number of metrics such as "average section length in characters" or "document length in words".
- The frequency of a small manually-created list of words often associated with sentiment including pronouns, obscenities and some simple emotive verbs eg. "think", "feel", "like", "hate".
- Non-word characters and character sequences such as punctuation and emoticons.
- Regex patterns to detect unusual word and punctuation structures. These include words containing 3 or more of the same character in a row ("arrrrgh"), excessive punction ("?!?!"), sequences of full stops ("....") and sequences of astrisks which might denote a censored word. For pattern or word counts, the vector items were max-min normalised.

2.2.2 *Syntactic Features*

We carried out experiments with a set of syntactic features in order to investigate whether knowledge of the relationships between the words in a sentence is useful in the opinion detection task. The text from the blog documents were parsed using the Charniak and Johnson re-ranking parser [2], a constituency parser which achieves state-of-the-art performance on the Wall Street Journal section of the Penn Treebank [6]. This parser also performs part-of-speech tagging on the text.

Parsing the blog entries for a retrieval run requires a significant amount of processing power. We used a high-end

computing cluster maintained by the Irish Centre for High End Computing. This cluster consists of 31 CPUs (AMD Opteron 250, 2.4 GHz single core). We only had time to parse our title only baseline run, so only runs based on that baseline make use of syntactic features.The marginal computing cost of parsing additional baselines is dependent on the amount of overlap between baselines.

Two kinds of features were extracted from the parsed data: part-of-speech n-grams and features related to the types of phrases occurring in the data:

- The 50 most discriminative part-of-speech unigrams, bigrams and trigrams were chosen, resulting in 150 part-of-speech n-gram features. The discriminativeness of an n-gram was determined by comparing the normalised count of the particular n-gram in the neutral dataset to the normalised count of the same n-gram in the opinionated dataset.
- The second set of features consisted of normalised document counts for each of the Penn Treebank phrasal types (*S*, *NP*, *ADJP*, etc.), for each of these phrasal types appearing as the root of a parse tree, and for miscellaneous parse tree structures which we thought might be more likely to reflect opinionated language, e.g. the number of occurrences of a subordinate clause within a verb phrase (e.g. *thought that...*) or an adverbial inside an adjectival phrase (e.g. *much more useful*).

2.2.3 Lexicon Features

Our lexicon-based features were based on aggregate scores derived from looking up words in the lexicon SentiWordNet [4]. SentiWordNet is a lexicon which assigns negative and positive scores to each sense of each synset in WordNet based on a semi-supervised classification of WordNet synsets. It has been used in the past for opinion finding by participants in the Blog Track to varying degrees of success [10], [12]. For our experiments, we considered the positive SentiWordNet score for a word w to be the mean of the positive scores for all the word senses of that word:

$$s_{pos}(w) = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{1}{m}\sum_{k=0}^{m} PosSwn_{i,k}\right) \quad (1)$$

where n is the number of synsets the word appears in, m is the number of word senses in the synset for that word and $PosSwn_{i,k}$ is the positivity score for word sense k in synset i for word w. The positive score for a document is the mean $s_{pos}(w)$ for all words in the document and is given by:

$$Score_{pos}(d) = \frac{1}{p}\sum_{i=0}^{p} s_{pos}(w_i) \quad (2)$$

for a document d with p words. The negative score is calculated similarly giving a feature vector of length two.

We also submitted runs where we did not use a classifier for lexicon scores. As the positive and negative values so closely resemble what the classifier score represents ("positivity", "negativity"), we thought it would be interesting to simply use the raw features as scores. For these runs we used the positive score from equation 2 for positive opinion retrieval runs and the negative score from the same equation for negative opinion retrieval runs. In this way, positive runs will not take into account negative scores and vice versa, as is the case with the classifier.

For opinion retrieval we used a weighted sum of the positive and negative scores to give an opinion score. Tuning the weights based on MAP from a combination of the 2006 and 2007 topics, we found a weighting of 0.7 for the positive score and 0.3 for the negative score to be a good approximate optimization. This might suggest that positive word orientation is more indicative of opinionate but could also simply reflect the larger volume of positive posts than negative posts in the training corpus. Whether each run uses a classifier for generating the lexicon module score is noted in Table 1 and Table 2.

2.3 Document Scoring

For the opinion retrieval task, we employed three logistic regression classifiers, one for each of the feature sets mentioned in Section 2.2. For the opinion retrieval task, each of the classifiers is a binary classifier which classifies documents as opinionated or non-opinionated. For the polarised opinion retrieval task we used the same system except the SentiWordNet classifier was retrained as a binary positive/non-positive classifier for positive blog retrieval and as a binary negative/non-negative classifier for negative blog retrieval. It was found during development that this performed significantly better than training either or both of the classifiers in the surface or syntactic module to detect polarity rather than opinion. This suggests that the linguistic characteristics do not differ as significantly between positive and negative blogs as between opinionated and non-opinionated. We found that using the surface and syntactic classifiers as prior subjectivity scores helped re-enforce the scores as designated by the polarity lexicon module.

2.3.1 Training

We used the qrels from the Blog Track in TREC 2006 and TREC 2007 as training for the classifiers in our system. Opinionated posts were considered to be those that were judged mixed, positive, or negative. Non-opinionated posts were considered to be those judged relevant. We considered negative posts to be those judged negative and non-negative posts were considered to be those judged mixed, relevant and positive. We considered positive posts to be those judged positive and non-positive posts were considered to be those judged mixed, relevant and negative. It is noted that the inclusion of mixed posts as non-relevant for polarised opinion retrieval is contentious as mixed posts by definition contain both negative and positive text.

2.3.2 Scoring

The relevant documents returned from Terrier are scored for opinion relevance in each of the modules as well as for negative and positive opinion in the lexicon classifier. The scores recorded are the probabilities from the probability distribution as determined by the logistic classifiers, except for the lexicon module which for some runs does not use a classifier.

2.4 Fusion

The scores in the system are fused through a multi-stage weighted CombSum late fusion [11]. For comparison we also submitted two unweighted fusion runs for the opinion retrieval run for baseline4. In the first stage, the scores from the three modules for a document d are added according to weights which sum to one:

$$Score_{op}(d) = w_1.s_{lex} + w_2.s_{surf} + w_3.s_{syn} \quad (3)$$

where:

$$w_1 + w_2 + w_3 = 1$$

For the runs based on baseline4 and our title and description baseline we omitted syntactic features so the fused opinion score becomes:

$$Score_{op}(d) = w_1.s_{lex} + w_2.s_{surf} \quad (4)$$

where:

$$w_1 + w_2 = 1$$

For polarity runs, the fused polarity score is calculated similarly, using the polarity scores from the lexicon module in place of the opinion lexicon score.
This score is then fused in a similar manner with the relevance score $Score_{rel}$ as determined by Terrier to give the overall relevant opinion score $Score_{relop}$:

$$Score_{relop}(d) = w_4.Score_{op} + (1 - w_4).s_{rel} \quad (5)$$

where:

$$w_4 <= 1$$

The weights we used for the runs we submitted are listed in Table 1 and Table 2.

These were obtained by grid searching in the parameter space $[w1, w2, w3, w4]$ ($[w1, w2, w4]$ for the runs without syntactic features) that produced optimizations of MAP for the combined 2006 and 2007 topics. It is interesting to note that whenever syntactic features are used, they appear to subsume the surface features, as optimizations result in the surface module weight tending towards 0.

The Terrier results for each topic are then re-ranked according to the combined relevance and opinion score $Score_{relop}$ to give the final ranking.

3. RESULTS AND ANALYSIS

The results for our opinion retrieval runs are detailed in Table 3. DCU's best run based on a non-distributed baseline was DCUCDVPtol. This run is based on our topic-only baseline and used syntactic features and a classifier-based lexicon module. This baseline performed better than the median on 29 out of the 50 new topics in 2008 as shown in Figure 2. The highest scoring opinion finding run we submitted based on a distributed baseline was DCUCDVPgoo which performs better than median on 44 out of the 50 new topics this year as shown in Figure 3. This run did not use syntactic features and used a classifier-based lexicon module.

Our results are consistent with the observations in last year's overview paper [5], that the higher the opinion MAP for the baseline, the more difficult it is to achieve a percentage increase above the baseline for opinion finding.

Table 3: Baseline Results for Opinion Retrieval

Run	Type	MAP	R-prec	P@10
DCUCDVPtbl	Title	0.2875	0.328	0.556
DCUCDVPtdbl	Title + Desc	0.264	0.3145	0.55
baseline4	Distributed	0.3822	0.4284	0.616

Table 4: Opinion Retrieval Results

Run	MAP	R-prec	P@10	% over b/l
DCUCDVPto	0.3216	**0.3706**	0.61	11.86
DCUCDVPtol	**0.3299**	0.3679	**0.636**	**14.75**
DCUCDVPtolnc	0.3296	0.3673	0.632	14.64
DCUCDVPtdo	0.2927	0.3439	0.594	10.87
DCUCDVPgo	0.4064	0.4392	0.676	6.33
DCUCDVPgonc	0.4052	0.4418	0.662	6.02
DCUCDVPgoo	**0.4155**	0.4479	**0.68**	**8.71**
DCUCDVPgoonc	0.4125	**0.4491**	0.674	7.93

In the graph, MAP is shown in descending order alongside the median MAP for that topic. For the run based on our topic-only baseline, there is a steady degradation in MAP and no sudden jumps or spikes in the graph. There is however a number of topics that perform significantly below baseline. On closer inspection, most of these topics also performed significantly worse that the median in the baseline run for topic retrieval. Some examples of low-performing topics include "System of a Down" and "I Walk the Line" which would have benefitted from word grouping or phrasing techniques. Such a pattern is not evident in the runs based on baseline4 where the retrieval baseline is much stronger and there is no topic where the median performs a significant amount better.

For polarised opinion detection our run based on our topic only baseline perfomed better than our run based on the topic and description baseline for both positive and negative opinion finding. This reflects a similar pattern to the relevance MAP results.

Our best polarity runs on baseline4 were for the unoptimized configuration for negative opinion finding and the optimized configuration for positive finding.

Our non-optimized fusion runs performed significantly worse than their optimized counterparts. Whether the lexicon module uses a classifier or not seems to have little effect. This is the only difference, for example, between DCUCDVPtol and DCUCDVPtolnc whose performance differs only by 0.0003 in terms of MAP.

For the polarity task our highest scoring run based on a non-distributed baseline was DCUCDVPtpl for both positive and negative opinion, which was based on our title only baseline. Regarding the distributed baseline, DCUCDVPgpo performed best for positive opinion, and DCUCDVPgp[2] for negative opinion. Comparing across the different baselines highlights surprising results — the runs based on

[2]DCUCDVPgp as listed in this paper differs from the of-

Table 1: Opinion Retrieval Configuration

Run	Baseline	Relevance Weight	Feature Weights			Classifier for Lexicon?
			Lexicon	Surface	Syntactic	
DCUCDVPto	Title	0.5	0.5	0.5	n/a	yes
DCUCDVPtol	Title	0.5	0.4	0	0.6	yes
DCUCDVPtolnc	Title	0.5	0.4	0	0.6	no
DCUCDVPtdo	Title + Description	0.5	0.5	0.5	n/a	yes
DCUCDVPgo	baseline4	n/a	n/a	n/a	n/a	yes
DCUCDVPgonc	baseline4	n/a	n/a	n/a	n/a	no
DCUCDVPgoo	baseline4	0.6	0.4	0.6	n/a	yes
DCUCDVPgoonc	baseline4	0.6	0.4	0.6	n/a	no

Table 2: Polarised Opinion Retrieval Configuration

Run	Polarity	Baseline	Relevance Weight	Feature Weights		
				Lexicon	Surface	Syntactic
DCUCDVPtpl	Title	positive	0.5	0.3	0.25	0.45
	Title	negative	0.5	0.6	0.05	0.35
DCUCDVPtdp	Title + Description	positive	0.35	0.35	0.65	n/a
	Title + Description	negative	0.5	0.65	0.35	n/a
DCUCDVPgp	baseline4	positive	n/a	n/a	n/a	n/a
	baseline4	negative	n/a	n/a	n/a	n/a
DCUCDVPgpo	baseline4	positive	0.5	0.35	0.65	n/a
	baseline4	negative	0.75	0.55	0.45	n/a

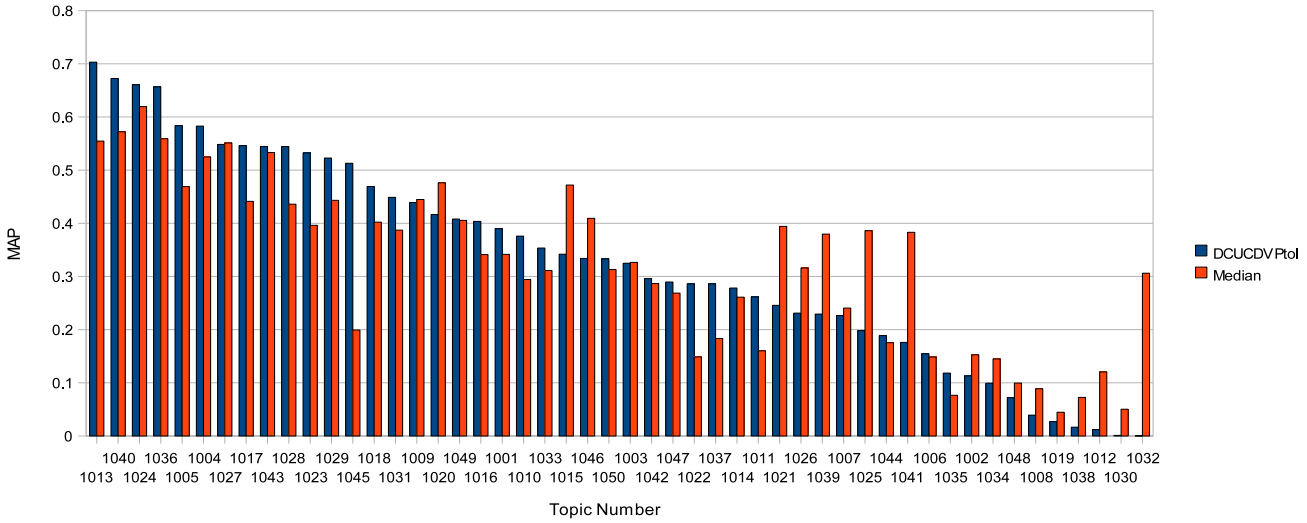

Figure 2: Per-topic MAP for the best-performing opinion run based on a non-distributed baseline, DCU's topic only baseline.

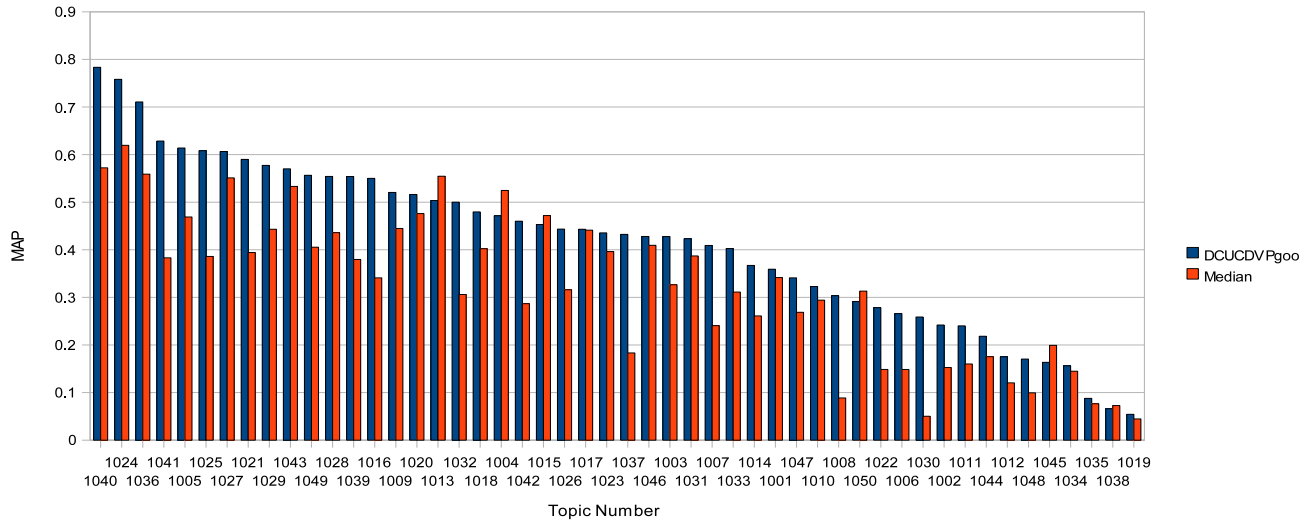

Figure 3: Per-topic MAP for DCU's best-performing opinion run based on a distributed baseline, baseline4.

Table 5: Polarised Opinion Retrieval Results

Run	Pol	MAP	R-prec	P@10	%
DCUCDVPtpl	pos	**0.1109**	0.1494	0.1408	**13.39**
	neg	**0.1111**	**0.1357**	**0.175**	**5.71**
DCUCDVPtdp	pos	0.1079	**0.1507**	**0.1469**	11.12
	neg	0.0932	0.1208	0.1417	5.19
DCUCDVPgp	pos	0.1642	**0.2074**	0.2028	5.05
	neg	**0.1528**	0.1714	0.1708	**19**
DCUCDVPgpo	pos	**0.1644**	0.1924	**0.2041**	**5.18**
	neg	0.1505	**0.1747**	**0.1813**	17.12

the distributed baseline performed much better for negative opinion finding and the runs based on our topic only baseline performed better on the positive opinion finding task in terms of percentage MAP above baseline. The reason for this is not entirely clear at this stage. It is worth noting that if the opinion finding runs are assessed for polarity, a similar pattern is observed.

4. CONCLUSIONS

We developed a system this year which evaluated three different approaches to opinion finding and polarised opinion finding. We also investigated the advantage in weighting different approaches in a fusion process.

We found that our most successful runs made use of syntactic features. Our best-performing runs also benefitted from using weighted scores when combining sources rather than unweighted combinations.

Inspection of our performance on a per topic basis shows that our opinion retrieval performance was hampered in several topics due to a low retrieval precision. Our runs based on baseline4, a much stronger baseline than our own, demonstrate a much more consistent performance.

ficially distributed results. The original submission was a based on an erroneous configuration.

It is also noted that subjectivity detection is a very important part of polarity detection and that with two of our three modules tuned to opinion finding, regardless of polarity, we achieved a significant improvement on baseline for the opinion finding task.

Acknowledgments

We would like to thank the Irish Centre for High-End Computing for the use of their computer facilities. We would also like to thank Joachim Wagner for the time and effort he spent parsing the document set. This work was supported by Science Foundation Ireland under grant number 07/CE/I1147 and by Enterprise Ireland Commercialisation Fund under grant number CFTD/2007/229.

5. REFERENCES

[1] http://htmlparser.sourceforge.net/.

[2] Eugene Charniak and Mark Johnson. Course-to-fine n-best-parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 173–180, Ann Arbor, Michigan, USA, June 2005.

[3] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03: Proceedings of the twelfth international conference on World Wide Web*, pages 519–528. ACM Press, 2003.

[4] Andrea Esuli and Fabrizio Sebastiani. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining.

[5] Craig MacDonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC-2007 Blog Track. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*, 2007.

[6] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[7] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. Terrier information retrieval platform. pages 517–519. 2005.

[8] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. Overview of the trec-2006 blog track. In *The Fifteenth Text REtrieval Conference (TREC 2007) Proceedings*, 2006.

[9] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up ?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[10] Song Rui, Tang Qin, Daming Shi, Hongfei Lin, and Zhihao Yang. DUTIR at TREC 2007 blog track. In *Proceedings of the Text Retrieval Conference (TREC)*, 2007.

[11] Joseph A. Shaw and Edward A. Fox. Combination of multiple searches. In *Proceeding of the 3rd Text Retrieval Conference (TREC-3)*, pages 105–108, 1995.

[12] Ethan Zhang and Yi Zhang. UCSC on TREC 2006 blog opinion mining. In *Proceedings of the Text Retrieval Conference (TREC)*, 2006.