Know-Center at TREC 2009 Blog Distillation Task: A Notebook Paper

Elisabeth Lex, Michael Granitzer, and Andreas Juffinger

Know-Center Graz, Inffeldgasse 21a, 8010 Graz, Austria
{elex,mgrani,ajuffinger}@know-center.at

Abstract. This paper outlines our experiments carried out at TREC 2009 Blog Distillation Task. Our system is based on a plain text index extracted from the XML feeds of the TREC Blogs08 dataset. This index was used to retrieve candidate blogs for the given topics. The resulting blogs were classified using a Support Vector Machine that was trained on a manually labelled subset of the TREC Blogs08 dataset. Three runs were conducted, one based on nouns, one based on stylometric properties, and one based on punctuation statistics. The facet identification based on our approach was successful, although a significant number of candidate blogs were not retrieved at all.

1 Introduction

On the web, a huge amount of blogs is available. However, the quality of these blogs is questionable in respect to quality because almost no mechanisms exist to validate the content before it is posted online [1].

In this year's TREC blog track, specifically the blog distillation task, exactly this problem is addressed. The goal of the track is to rank candidate blogs that are relevant to a topic or query and to assign the retrieved blogs to a set of facets that represent different quality aspects [1].

We consider this as a two-step procedure: (i) the first step is to retrieve blogs that are relevant to a topic or query, respectively. This is an Information Retrieval (IR) process. The second step (ii) is to assign particular facets of interest to the retrieved blogs. For the second step, we applied supervised classification on a manually labelled subset of the TREC Blogs08 dataset. Supervised classification has been successfully applied to text classification [2] and facet classification like for instance emotion classification [3] . We trained a classifier on a subset of blogs randomly taken from the TREC Blogs08 dataset [2] and labelled them manually according to the facets. The classifier is then used to categorise the blogs into the facets.

The rest of this paper is structured as follows: Section 2 outlines the collection and the preprocessing of the data, Section 3 describes the methods and runs, in Section 4, the results are given, and Section 5 concludes our work.

[1] http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG
[2] http://ir.dcs.gla.ac.uk/test_collections/blogs08info.html

2 Collection and Preprocessing

The experiments for the blog distillation task are carried out on the Blogs08 dataset. The Blogs08 dataset has 28.488.767 blog posts from 1.303.520 blog feeds [3]. It samples the blogosphere from January 2008 to February 2009.

For the first step, the retrieval task, we vectorised and indexed 606.939 different blogs out of 1.303.520 blogs. From our indexed blogs, we retrieved 3.476 million different blog feeds. Our search index size was 41 GB, and the size of the whole repository with vectors for all features was 220 GB.

Previous experiences have shown that an analysis has to be performed on blog post level [6]. To extract the blog posts, we exploited the permalinks.

For the second step, the classification task, we manually annotated 83 blogs by the facets given in the TREC blog task. These 83 blogs were selected randomly to guarantee an independent sample. Also, due to the random selection, we assume that the label distribution reflects the real blog world although the number of blogs is not high. As mentioned before, we assigned the facets on blog post level which resulted in 12870 annotated blog posts. We then transformed the unstructured blog posts in a structured form and thus prepared the blog posts for our text mining unit.

Note that we first filtered the blogs by language in order to retrieve only English blogs. For this, we exploited a language guesser developed at our institution that is based on n-grams and the Apache Nutch project [4].

From the remaining structured blog posts, we extracted nouns, sentences, and punctuation features. For each set of features, we created a vector and stored these vectors in our index. Our search framework is based on Apache Lucene which is an open source search engine that is able to index several gigabytes of documents at reasonable time. Also, complex searches can be performed quickly on the documents. Besides, Lucene maintains a relevance ranking scoring which is applicable to find the best matches to queries. We used this relevance score for ranking, as described later.

To sum up, our resulting training set consists of 12870 blog posts. We trained our classifier on the annotated blog posts and also classified the blogs on blog post level. To assign the facets to the whole blogs, we performed a majority voting over the posts.

2.1 Relevance Ranking

In order to derive a final relevance ranking, we normalised the Lucene score so that it lies between 0 and 1 by dividing it by the maximum score which is shown in Equation 1.

$$luceneScore = score/maxScore \tag{1}$$

[3] http://ir.dcs.gla.ac.uk/test_collections/blogs08info.html
[4] http://lucene.apache.org/nutch/

Then, we calculated the final relevance score whereas we took into consideration both the Lucene score and the classifier confidence for the particular facet. The formula for the final score is shown in Equation 2.

$$finalScore = \alpha * lucenceScore + (1 - \alpha) * facetConfidence \quad (2)$$

3 Methods and Runs

Four the retrieval task, we used Apache Lucene as search engine, as mentioned before. With Lucene, we searched our blog index for the 50 topics and ranked and sorted the top 100 blogs according to our relevance ranking scheme (see Equation 2). We then classified the top 100 blogs into the given facets using the trained classifier. As a classification algorithm, we applied a Support Vector Machine based on LibLinear [4] with standard parametrisation.

We carried out three runs whereas each run is characterised by the features used in this run:

1. *Nounfull NF*: For this run, we used a feature space only with nouns annotated by the OpenNLP library [5].
2. *Punctfull PF*: In this run, we created a feature space only with punctuation features. Contains the punctuation distribution, the count of one of 12 punctuations per documents (12 dimensions).
3. *Sentencefull SF*: For this run, we used a feature space with stylometric properties based on sentence statistics.

Clearly, a number of different features can be used for text classification problems, as described in [7]. In our feature selection strategy, we considered both lexical and stylometric features. Nouns belong to the category of lexical features and generally cover topics. Usually, a classification based on nouns yields good results, at least in our experience. Punctuation features and features based on sentence statistics are stylometric features. In general, stylistic variations are described by stylometric and therefore such features are often used for authorship identification [5]. In our experiments we applied stylometric features to guarantee topic-independence. Our goal was to avoid that the classifier learns topics instead of characteristics.

4 Results

With our methodology, we achieved the following results:

Of the 50 topics, we were able to retrieve 39 topic which had at least one relevant blog for each side of the facet (e.g., one relevant opinionated blog and one relevant factual blog). Therefore, all our results relate to these 39 topics. Note that we did not manage to vectorize and index the whole Blogs08 dataset.

[5] http://opennlp.sourceforge.net

Therefore, relevant blogs were missing in our index which is reflected in poor results for certain topics.

In Table 1, the results for all runs are summarised. For each run, three rankings of the blogs were performed. The first, "none", denotes a ranking with no facet value applied. The second "first" correspond to the first value of a facet ("opiniated", "in-depth", "personal") and the third "second" represent the second value of a facet ("factual", "shallow", "official").

The ranking "none" can be understood as a baseline ranking.

RUN	Facet	MAP	R-Precision	P@10
NF	none	0.0624	0.0980	0.1410
NF	first	0.0538	0.0725	0.0769
NF	second	0.0231	0.0346	0.0256
PF	none	0.0624	0.0980	0.1410
PF	first	0.0691	0.0846	0.0692
PF	second	0.0227	0.0339	0.0308
SF	none	0.0624	0.0980	0.1410
SF	first	0.0559	0.0717	0.0667
SF	second	0.0302	0.0334	0.0436

Table 1. Results for all facets for all runs

From Table 1 one can see that the obtained MAP values for all rankings are quite low. In general, MAP corresponds to the average precision and emphasises returning more relevant documents earlier. Due to the fact that we were only able to index less than half of the data, our retrieval was suboptimal. More specifically, in many cases we did not retrieve enough relevant blogs for a topic and this lead to quite low MAP values.

For instance, in run NF with the second facet enabled, only 69 blogs out of 331 relevant blogs were retrieved and consequently, the MAP value for this run was low, as shown in Table 2.

	Results
numQueries	39
numRetrieved	3801
numRelevant	331
numRelevantRetrieved	69
map	0.0231

Table 2. NF: Results for second facet - all

In cases where our retrieval was successful which means we obtained enough relevant blogs for a topic, also good MAP results were achieved. An example of this is outlined in Table 3, which holds the results for the first facet for topic 1101:

TOPIC	RUN	MAP	R-Precision	P@10
1101	NF	0.2590	0.4000	0.5000
1101	PF	0.2989	0.4571	0.6000
1101	SF	0.2221	0.4143	0.3000

Table 3. SENTENCE: Results for first facet for topic 1101

Another example where the classification was successful was topic 1103, as shown in Table 4:

TOPIC	RUN	MAP	R-Precision	P@10
1103	NF	0.0525	0.1429	0.1000
1103	PF	0.2857	0.2857	0.2000
1103	SF	0.2381	0.2857	0.2000

Table 4. SENTENCE: Results for first facet for topic 1103

Compared to the summary statistics for the first facet "opinionated" , which is given in Table 5, one can see, that for this topic 1103, our results were above average for the runs "punctfull" and "sentencefull".

MAP1.	MAP2	MAP3	R-Prec1.	R-Prec2
0.4286	0.1667	0.0000	0.4286	0.2857

Table 5. SENTENCE: Results for first facet for topic 1103

In some cases, our relevance ranking was not optimal as is reflected in low values for R-precision.

In later experiments on the Blogs08 dataset where we also used the annotated blogs for training, we were able to achieve a classification accuracy of 0.75 for nouns only. In other experiments on other features e.g. stems, we reached an accuracy of 0.91 on blog post level. Summing up, if we improve our retrieval and ranking, we can definitely improve our results. Also, later experiments revealed that if we use features like stems, an improvement of 15-30% of the classification accuracy can be achieved.

5 Conclusions

In this work, we described the efforts and results for our participation in the TREC blog track 2009. Our system is based on a plain text index extracted from the XML feeds only. We successfully indexed 680k of 1.3 Mio blogs and this index was used to retrieve candidate blogs for the given topics. From the top 2500 result blog entries, the top 100 blogs were identified according to the accumulated

relevance score of the particular blog entries. The resulting blogs were classified using a Support Vector Machine trained on a manually labelled subset of the TREC Blogs08 dataset. Three runs were conducted, one based on nouns, one based on stylometric properties, and one based on punctuation statistics. The facet identification based on this approach was successful, although a significant number of candidate blogs were not retrieved at all.

6 Acknowledgements

The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria.

References

1. C. C.-D. D. G. A. Agichtein, E. and G. Mishne. Findin high-quality content in social media with application to community-based question answering. In *Proceedings of WSDM*, 2008.
2. A. An, L. Viii, L. Viii, L. Viii, T. Joachims, T. Joachims, F. Informatik, F. Informatik, F. Informatik, F. Informatik, and F. Informatik. Text categorization with support vector machines: Learning with many relevant features, 1998.
3. H.-H. C. Changhua Yang, Kevin Hsin-Yih Lin. Emotion classification using web blog corpora. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 2007.
4. R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 2008.
5. J. Grieve. Quantitative authorship attribution: An evaluation of techniques. *Literary and Linguistic Computing*, 2007.
6. A. Juffinger, M. Granitzer, and E. Lex. Blog credibility ranking by exploiting verified content. In *WICOW '09: Proceedings of the 3rd workshop on Information credibility on the web*, pages 51–58. ACM, 2009.
7. J. Karlgren and J. Cutting. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of COLING*, pages 1071–1075, 1994.