Feedback Prediction for Blogs

Krisztian Buza

Budapest University of Technology and Economics
Department of Computer Science and Information Theory buza@cs.bme.hu

Abstract. The last decade lead to an unbelievable growth of the importance of social media. Due to the huge amounts of documents appearing in social media, there is an enormous need for the *automatic* analysis of such documents. In this work, we focus on the analysis of documents appearing in blogs. We present a proof-of-concept industrial application, developed in cooperation with Capgemini Magyarország Kft. The most interesting component of this software prototype allows to predict the number of feedbacks that a blog document is expected to receive. For the prediction, we used various predictions algorithms in our experiments. For these experiments, we crawled blog documents from the internet. As an additional contribution, we published our dataset in order to motivate research in this field of growing interest.

1 Introduction

The last decade lead to an unbelievable growth of the importance of social media. While in the early days of social media, blogs, tweets, facebook, youtube, social tagging systems, etc. served more-less just as an entertainment of a few enthusiastic users, nowadays news spreading over social media may govern the most important changes of our society, such as the revolutions in the Islamic world, or US president elections. Also advertisements and news about new products, services and companies are spreading quickly through the channels of social media. On the one hand, this might be a great possibility for promoting new products and services. On the other hand, however, according to sociological studies, negative opinions spread much quicker than positive ones, therefore, if negative news appear in social media about a company, the company might have to react quickly, in order to avoid losses.

Due to the huge amounts of documents appearing in social media, analysis of all these documents by human experts is hopeless, and therefore there is an enormous need for the *automatic* analysis of such documents. For the analysis, however, we have to take some special properties of the application domain into account. In particular, the uncontrolled, dynamic and rapidly-changing content of social media documents: e.g. when a blog-entry appears, users may immediately comment this document.

We developed a software prototype in order to demonstrate how data mining techniques can address the aforementioned challenges. This prototype has the following major components: (i) the crawler, (ii) information extractors, (iii) data store and (iv) analytic components. In this paper, we focus on the analytic components that allow to predict the number of feedbacks that a document is expected to receive in the next 24 hours. For feedback prediction, we focused on the documents appearing in blogs and performed experiments with various predictions models. For these experiments we crawled Hungarian blog sites. As an additional contribution, we published our data.

2 Related Work

Data mining techniques for social media have been studied by many researchers, see e.g. (Reuter et al. 2011) and (Marinho et al. 2008). Our problem is inherently related to many web mining problems, such as opinion mining or topic tracking in blogs. For an excellent survey on opinion mining we refer to (Pang and Lee 2008). Out of the works related to blogs we point out that Pinto (2008) applied topic tracking methods, while Mishne (2007) exploited special properties of blogs in order to improve retrieval.

Despite its relevance, there are just a few works on predicting the number of feedbacks that a blog-document is expected to receive. Most closely related to our work is the paper of Yano and Smith (2010) who used Naive Bayes, Linear and Elastic Regression and Topic-Poisson models to predict the number of feedbacks in political blogs. In contrast to them, we target various topics (do not focus on political blogs) and perform experiments with a larger variety of models including Neural Networks, RBF Networks, Regression Trees and Nearest Neighbor models.

3 Domain-specific concepts

In order to address the problem, first, we defined some domain-specific concepts that are introduced in this Chapter. We say that a *source* produces *documents*. For example, on the site *torokgaborelemez.blog.hu*, new documents appear regularly, therefore, we say that torokgaborelemez.blog.hu is the source of these documents.

From the point of view of our work, the following parts of the documents are the most relevant ones: (i) *main text of the document:* the text that is written by the author of the document, this text describes the topic of the document, (ii) *links to other documents:* pointers to semantically related documents, in our case, trackbacks are regarded as such links, (iii) *feedbacks:* opinions of social media users about a document is very often expressed in form of feedbacks that the document receives. Feedbacks are usually short textual comments referring to the main text of the document and/or other feedbacks. – Temporal aspects of all the above entities are relevant for our task. Therefore, we extract time-stamps for the above entities and store the data together with these timestamps.

4 Feedback prediction

Feedback prediction is the scientifically most interesting component of the prototype, therefore we focus on feedback prediction. For the other components of the software prototype we refer to the presentation slides available at http://www.cs.bme.hu/~buza/pdfs/gfkl_buza_social_media.pdf .

4.1 Problem Formulation

Given some blog documents that appeared in the past, for which we already know when and how many feedbacks they received, the task is to predict how many feedbacks *recently* published blog-entries will receive in the next H hours. We regard the blog documents published in the last 72 hours as recently published ones, we set $H = 24$ hours.

4.2 Machine Learning for Feedback Prediction

We address the above prediction problem by machine learning, in particular by regression models. In our case, the instances are the recently published blog documents and the target is the number of feedbacks that the blog-entry will receive in the next H hours.

Most regression algorithms assume that the instances are vectors. Furthermore, it is assumed that the value of the target is known for some (sufficiently enough) instances, and based on this information, we want to predict the value of the target for those cases where it is unknown. First, using the cases where the target is known, a prediction model, *regressor*, is constructed. Then, the regressor is used to predict the value of the target for the instances with unknown valued target.

In our prototype we used neural networks (multilayer perceptrons in particular), RBF-networks, regression trees (REP-tree, M5P-tree), nearest neighbor models, multivariate linear regression and bagging out of the ensemble models. For more detailed descriptions of these models we refer to (Witten and Frank 2005) and (Tan et al. 2006).

In the light of the above discussion, in order to apply machine learning to the feedback prediction problem, we have to resolve two issues: (i) we have to transform the instances (blog documents) into vectors, and (ii) we need some data for which the value of the target is already known (train data).

For the first issue, i.e., for turning the documents into vectors, we extract the following features from each document:

1. basic features: number of links and feedbacks in the previous 24 hours relative to baseTime; number of links and feedbacks in the time interval from 48 hours prior to baseTime to 24 hours prior to baseTime; how the number of links and feedbacks increased/decreased in the past (the past is seen relative to baseTime); number of links and feedbacks in the first 24 hours after the publication of the document, but before baseTime; aggregation of the above features by source,
2. textual features: the most discriminative bag of words features,[1]
3. weekday features: binary indicator features that describe on which day of the week the main text of the document was published and for which day of the week the prediction has to be calculated,
4. parent features: we consider a document d_P as a patent of document d, if d is a reply to d_P, i.e., there is a trackback link on d_P that points to d; parent features are the number of parents, minimum, maximum and average number of feedbacks that the parents received.

We solve the first issue as follows: we select some date and time in the past and simulate as if the current date and time would be the selected date and time. We call the selected date and time *baseTime*. As we actually know what happened after the baseTime, i.e., we know how many feedbacks the blog entries received in the next H hours after *baseTime*, we know the values of the target for these cases. While doing so, we only take blog pages into account that were published in the last three days relative to the baseTime, because older blog pages usually do not receive any more new feedbacks.

A similar approach allows us to quantitatively evaluate the prediction models: we choose a time interval, in which we select different times as baseTime, calculate the value of the target and use the resulting data to train the regressor. Then, we select a disjoint time interval in which we again take several baseTimes and calculate the true values of the target. However, the true values of the target remain hidden for the prediction model, we use the prediction model to estimate the values of the targets for the second time interval. Then we can compare the true and the predicted values of the target.

5 Experiments

We examined various regression models for the blog feedback prediction problem, as well as the effect of different type of features. The experiments, in total, took several months of CPU time into account.

[1] In order to quantify how discriminative is a word w, we use the average and standard deviation of the number of feedbacks of documents that contain w, and the average and standard deviation of the number of feedbacks of documents that *do not* contain w. Then, we divide the difference of the number of average feedbacks with the sum of the both standard deviations. Then, we selected the 200 most discriminative words.

5.1 Experimental Settings

We crawled Hungarian blog sites: in total we downloaded 37279 pages from roughly 1200 sources. This collection corresponds approximately 6 GB of plain HTML document (i.e., without images). We preprocessed as described in Section 4.2. The preprocessed data had in total 280 features (without the target variable, i.e., number of feedbacks). In order to assist reproducibility of our results as well as to motivate research on the feedback prediction problem, we made the preprocessed data publicly available at http://www.cs.bme.hu/~buza/blogdata.zip .

In the experiments we aimed to simulate the real-world scenario in which we train the prediction model using the blog documents of the past in order to make predictions for the blog documents of the present, i.e., for the blog documents that have been published recently. Therefore, we used a temporal split of the train and test data: we used the blog documents from 2010 and 2011 as train data and the blog documents from February and March 2012 as test data. In both time intervals we considered each day as baseTime in the sense of Section 4.2.

For each day of the test data we consider 10 blog pages that were *predicted* to have to largest number of feedbacks. We count how many out of these pages are among the 10 pages that received the largest number of feedbacks *in the reality.* We call this evaluation measure *Hits@10* and we average Hits@10 for all the days of the test data.

For the AUC, i.e., area under the receiver-operator curve, see (Tan et al., 2006), we considered as positive the 10 blog pages receiving the highest number of feedbacks *in the reality.* Then, we ranked the pages according to their *predicted* number of feedbacks and calculated AUC. We call this evaluation measure *AUC@10.*

For the experiments we aimed at selecting a representative set of state-of-the-art regressors. Therefore, we used multilayer perceptrons (MLP), linear regressors, RBF-Networks, REP-Trees and M5P-Trees. These regressors are based on various theoretical background (see e.g. neural networks versus regression trees). We used the Weka-implementations of these regressors, see (Witten and Frank 2005) for more details.

5.2 Results and Discussion

The performance of the examined models, for the case of using all the available features is shown in Figure 1. For MLP, we used a feed-forward structure with (i) 3 hidden neurons and one hidden layer and (ii) 20 and 5 hidden neurons in the first and second hidden layers. In both cases we set the number of training iteration of the Backpropagation Algorithm to 100, the learning rate to 0.1 and the momentum to 0.01. For the RBF-Network, we tried various number of clusters, but they did not have substantial impact on the results. We present results for 100 clusters.

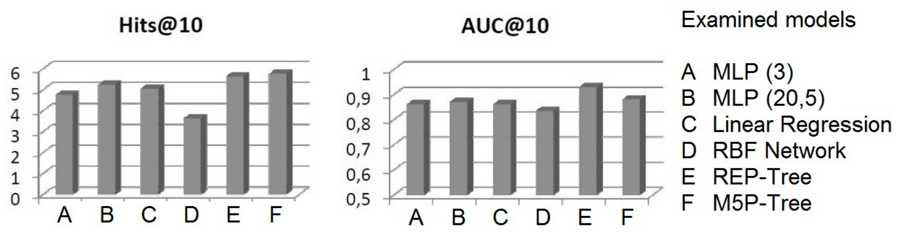

Fig. 1. The performance of the examined models.

Table 1. The effect of different types of features and the effect of bagging. The performance (Hits@10 and AUC@10) of the models for different feature sets.

Model	Basic	Basic + Weekday	Basic + Parent	Basic + Textual	Bagging
MLP (3)	5.533 ± 1.384	5.550 ± 1.384	5.612 ± 1.380	4.617 ± 1.474	5.467 ± 1.310
	0.886 ± 0.084	0.884 ± 0.071	0.894 ± 0.062	0.846 ± 0.084	0.890 ± 0.080
MLP (20,5)	5.450 ± 1.322	5.488 ± 1.323	5.383 ± 1.292	5.333 ± 1.386	5.633 ± 1.316
	0.900 ± 0.080	0.910 ± 0.056	0.914 ± 0.056	0.896 ± 0.069	0.903 ± 0.069
k-NN	5.433 ± 1.160	5.083 ± 1.345	5.400 ± 1.172	3.933 ± 1.223	5.450 ± 1.102
($k = 20$)	0.913 ± 0.051	0.897 ± 0.061	0.911 ± 0.052	0.850 ± 0.060	0.915 ± 0.051
RBF Net	4.200 ± 1.458	4.083 ± 1.320	3.414 ± 1.700	3.833 ± 1.428	4.750 ± 1.233
(clusters: 100)	0.860 ± 0.070	0.842 ± 0.069	0.846 ± 0.074	0.818 ± 0.074	0.891 ± 0.050
Linear	5.283 ± 1.392	5.217 ± 1.343	5.283 ± 1.392	5.083 ± 1.215	5.150 ± 1.327
Regression	0.876 ± 0.088	0.869 ± 0.097	0.875 ± 0.091	0.864 ± 0.096	0.881 ± 0.082
REP Tree	5.767 ± 1.359	5.583 ± 1.531	5.683 ± 1.420	5.783 ± 1.507	5.850 ± 1.302
	0.936 ± 0.038	0.931 ± 0.042	0.932 ± 0.043	0.902 ± 0.086	0.934 ± 0.039
M5P Tree	6.133 ± 1.322	6.200 ± 1.301	$6,000 \pm 1.342$	6.067 ± 1.289	5.783 ± 1.305
	0.914 ± 0.073	0.907 ± 0.084	0.913 ± 0.081	0.914 ± 0.068	0.926 ± 0.048

The effect of different feature types and the effect of bagging is shown in Table 1. For bagging, we constructed 100 randomly selected subsets of the basic features and we constructed regressors for all of these 100 subsets of features. We considered the average of the predictions of these 100 regressors as the prediction of the bagging-based model.

The number of hits was around 5-6 for the examined models, which was much better than the prediction of a naive model, i.e., of a model that simply predicts the average number of feedbacks per source. This naive model achieved only 2-3 hits in our experiments. In general, relatively simple models, such as M5P Trees and REP Trees, seem to work very well both in terms of quality and runtime required for training of these models and for prediction using these models. Depending on the parameters of neural networks, the training may take relatively long time into account. From the quality point of view, while we observed neural networks to be competitive to the regression trees, the examined neural networks did not produce much better results than the mentioned regression trees.

Additionally to the presented results, we also experimented with support vector machines. We used the Weka-implementation of SVM, which had inacceptably long training times, even in case of simple (linear) kernel.

Out of the different types of features, the basic features (including aggregated features by source) seem to be the most predictive ones.

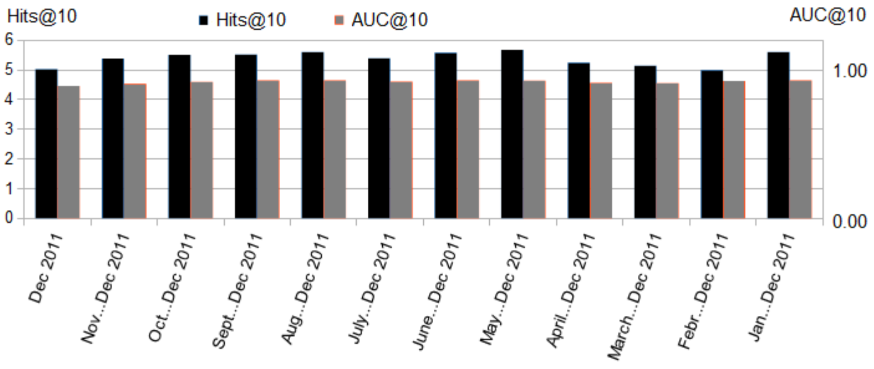

Fig. 2. The performance of the REP-tree classifier with basic features for various training intervals.

Bagging, see the last column of Table 1, improved the performance of MLPs and RBF-Network both in terms of Hits@10 and AUC@10, and the performance of REP-tree in terms of Hits@10. In the light of average and standard deviation, these improvement are, however, not significant.

We also examined how the length of the training interval affects the quality of prediction: both Hits@10 and AUC@10 of the REP-tree classifier are shown in Figure 2 for various training intervals. As expected, recent training intervals, such as the last one or two months of 2011, seem to be informative enough for relatively good predictions. On the other hand, with using more and more historical data from larger time intervals, we did not observe a clear trend which may indicate that the user's behavior may (slightly) change and therefore historical data of a long time interval is not necessary more useful than recent data from a relatively short time interval.

6 Conclusion

In the last decade, the importance of social media grew unbelievably. Here, we presented a proof-of-concept industrial application of social media analysis. In particular, we aimed to predict the number of feedbacks that blog documents receive. Our software prototype allowed to crawl data and perform experiments. The results show that state-of-the art regression models perform well, they outperform naive models substantially. We mention that our partners at Capgemini Magyarország Kft. were very satisfied with the results. On the other hand, the results show that there is room for improvement, while developing new models for the blog feedback prediction problem seems to be a non-trivial task: with widely-used techniques, in particular ensemble methods, we only achieved marginal improvement. In order to motivate research in this area of growing interest, we made our data publicly available.

Acknowledgment. We thank Capgemini Magyarország Kft. for the financial support of the project. The work reported in the paper has been developed in the framework of the project ”Talent care and cultivation in the scientific workshops of BME” project. This project is supported by the grant TÁMOP-4.2.2.B-10/1–2010-0009.

References

Marinho LB, Buza K, Schmidt-Thieme L (2008) Folksonomy-Based Collabulary Learning The Semantic Web - ISWC 2008, LNCS, 5318, 261-276

Mishne G (2007) Using Blog Properties to Improve Retrieval. International Conference on Weblogs and Social Media

Pang B, Lee L (2008) Opinion Mining and Sentiment Analysis *Journal Foundations and Trends in Information Retrieval, 2, 1-135.*

Pinto JPGS (2008) Detection Methods for Blog Trends. Report of Dissertation Master in Informatics and Computing Engineering, Faculdade de Engenharia da Universidade do Porto

Reuter T, Cimiano P, Drumond L, Buza K, Schmidt-Thieme L (2011) Scalable Event-Based Clustering of Social Media Via Record Linkage Techniques, 5th International AAAI Conference on Weblogs and Social Media

Tan PN, Steinbach M, Kumar V (2006) Introduction to Data Mining. Pearson Addison Wesley.

Witten IH, Franke E (2005) Data Mining. Practical Machine Learning Tools and Techniques. Elsevier, Morgan Kaufmann, second edition.

Yano T, Smith NA (2010) Whats Worthy of Comment? Content and Comment Volume in Political Blogs. *4th International AAAI Conference on Weblogs and Social Media, 359–362*