

## Analyzing Political Trends in the Blogosphere

Gianluca Demartini\*, Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl

L3S Research Center – Appelstr. 9a 30167 Hannover, Germany  
 {demartini,siersdorfer,chelaru,nejdl}@L3S.de

### Abstract

In the last years, the blogosphere has become a vital part of the web, covering a variety of different points of view and opinions on political and event-related topics such as immigration, election campaigns, or economic developments. Tracking the public opinion is usually done by conducting surveys resulting in significant costs both for interviewers and persons consulted. In this paper, we propose a method for extracting political trends in the blogosphere. To this end, we apply sentiment and time series analysis techniques in combination with aggregation methods on blog data to estimate the temporal development of opinions on politicians.

### Introduction

The blogosphere has attracted an active web community in the recent years and has become a popular forum for sharing opinions and thoughts on a variety of issues. Topics discussed range from rather casual themes, such as sports, concerts, or celebrities to more complex and polarizing political ones such as abortion, elections, or immigration. Blog data constitutes a powerful source for mining information about opinions and trends (Macdonald et al. 2010).

Public opinion on different topics is usually estimated by professional services conducting surveys on a sample of the population. For instance, opinions about political elections are estimated by interviewing electors on the phone. This is clearly an expensive activity for both the company carrying out the interviews as well as for the sampled electors who have to spend their time answering questions.

In this paper, we propose an approach towards overcoming the drawbacks of opinion polls, and present models for automatically estimating public opinions from the blogosphere by mining and aggregating information extracted from blogs over time. More specifically, we define a pipeline of several IR and NLP components to determine the public opinion towards the two US 2008 presidential candidates, Obama and McCain. Figure 1 depicts an example comparing our automatically computed estimate (exclusively based on blog data) to traditional opinion polls about the main rival candidates Obama and McCain over time, illustrating how blogs are reflected in the real world.

\*This work is partially supported by the EU Large Scale Integrated Project LivingKnowledge (contract no. 231126). Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

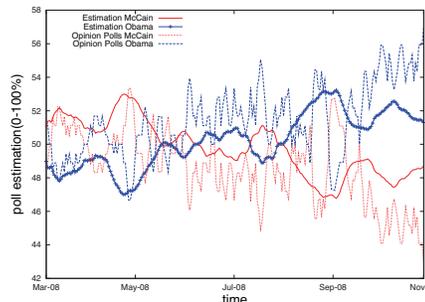


Figure 1: Temporal development of public opinion for US election candidates Obama and McCain in 2008 (estimates from the blogosphere and vis-a-vis traditional poll results).

### Related Work

Blog mining has primarily been studied in the context of the Text REtrieval Conference (TREC) Blog Track since 2006. The main tasks are about retrieving opinionated postings and their polarity, and retrieving topical items both at posting and feed level. In order to retrieve opinions most approaches adopt a two-stage process. First, documents are ranked according to topical relevance, using mostly off-the-shelf retrieval systems and weighting models. Results are then re-ranked or filtered by applying one or more heuristics for detecting opinions. In contrast to TREC Blog tasks and previous work we are using the described techniques rather as preprocessing steps towards providing an aggregated and time-aware view on opinions.

Related to our paper is the work (Liu et al. 2007) in the context of sales prediction where the authors predict movie incomes by mining blogs, and combine latent analysis techniques for determining sentiment. In contrast to our work, the approach always requires the availability of training samples, and predictions are made just for a very short time period (typically around 10 days). Recently, (O'Connor et al. 2010) studied the aggregation of opinions on Twitter data, using more simplistic methods for aggregating sentiment over a sample of tweets. However, specifically for the challenging US election scenario, they report a very low correlation between their estimates on Twitter and traditional polls, and, thus, are not able to capture trends.

### Extracting opinions from blogs

In this section we present our methods for extracting bloggers' opinions on two target entities ( $O$  and  $M$ ) and aggre-

gating them over time. The approach we propose for estimating the public opinion about politicians consists of three main steps: 1) retrieval, 2) sentiment classification, and 3) aggregation.

## Estimating Opinion Polls

**Retrieving Relevant Postings.** From the blogosphere we first retrieve a set  $P = \{p_1, \dots, p_n\}$  of blog postings relevant to entities  $O$  and  $M$ . In the Obama vs. McCain example we were conducting this based on simple keyword search, assuming that all postings containing at least one of the candidates names were potentially useful for further analysis.

**Assigning Sentiment Scores.** We aim to assign sentiment values  $s(p)$  to blog postings  $p$ . For our scenario with two concurrent entities  $O$  and  $M$ , we assume that  $s(p)$  lies in the interval  $[-1, +1]$  with  $s(p) = 1$  referring to a maximum positive opinion expressed about  $O$ , and, conversely  $s(p) = -1$  corresponding to a totally positive opinion on the competing entity  $M$ .

The first option to obtain these sentiment values is to exploit a lexical resource for opinion mining such as SentiWordNet (Esuli and Sebastiani 2006) built on top of WordNet (Fellbaum 1998). In SentiWordNet a triple of three *sentiment values* (*pos, neg, obj*) (corresponding to positive, negative, or rather neutral sentiment flavor of a word respectively) are assigned to each set of synonymous words  $w$  in WordNet. We define  $Sent_O(p)$  as the set of sentences in a posting  $p$  which contains entity  $O$  but not  $M$  and we analogously define  $Sent_M(p)$ . Let  $Lex(st)$  be the set of words  $w$  from sentence  $st$  that can be assigned a sentiment value based on the available lexicon. For a sentence  $st$  we aggregate positive and negative sentiment values for opinionated words as follows:  $senti_e(st) = \frac{\sum_{w \in Lex(st)} pos(w) - neg(w)}{|Lex(st)|}$ ,  $e \in \{O, M\}$  where,  $pos(w)$  and  $neg(w)$  are the positive / negative values for the word  $w$  in the lexical resource, and  $st \in Sent_e(p)$ . We can now compute the sentiment of the posting  $p$  about each of the two considered entities independently. Thus, we define:  $s_e(p) = \frac{\sum_{st \in Sent_e(p)} senti_e(st)}{|Sent_e(p)|}$ ,  $e \in \{O, M\}$ . The values of these estimators lie in the interval  $[-1, +1]$  where  $-1$  indicates a strong negative opinion and  $+1$  a strong positive opinion about the candidate  $e$ . Then, we compute the overall preference estimate for posting  $p$  as  $s(p) = \frac{s_O(p) - s_M(p)}{2}$ . The estimator  $s(p)$  assumes values in  $[-1, +1]$  where  $-1$  indicates a preference towards  $M$  and  $+1$  a preference for  $O$ .

A second option for computing the sentiment of bloggers with respect to given entities is the application of text classification on a sentence level. We use a standard Bag-of-Words representation of sentences with stemming, stopword removal, and TF weighting. An SVM classifier is trained on opinionated data from different sources to ensure a diverse coverage: manual judgments from the TREC 2008 Blog Track (Ounis, Macdonald, and Soboroff 2008) consisting of 18,142 opinionated blog postings, 100,000 reviews crawled from the website *epinions.com* assigned to a positive or negative class considering the 1 or 5 stars judgement associated with the textual review, and a sentence polarity dataset (Pang and Lee 2005).

For SVMs a natural confidence measure is the distance of a test sentence vector from the separating hyperplane. After linearly normalizing classification scores for a sentence  $st$  to be in the range  $[-1, 1]$ , we compute the sentiment values for posting  $p$  with respect to entities  $O$  and  $M$ . These values are then combined to the preference estimate  $s(p)$ .

**Sentiment Aggregation.** For a time interval  $t = [t_1, t_2]$  we consider all postings  $P_t = \{p \in P : t_1 \leq TS(p) \leq t_2\}$  relevant to entities  $O$  and  $M$  published in that interval, where  $TS(p)$  is the timestamp of posting  $p$ . We then use data extracted from this set to estimate an opinion value  $poll(t) \in [-1, +1]$  for time interval  $t$ . Given the preference estimates  $s(p)$  for postings we want to estimate  $poll(t)$  through aggregation. Let  $Sel$  be a function for selecting the subset  $Sel(P_t) = P'_t \subseteq P_t$  we want to consider for aggregation, and  $f : [-1, +1] \rightarrow [-1, +1]$  be a monotonically increasing function with  $f(-1) = -1$  and  $f(1) = 1$ . Then, we can compute an aggregated value as follows:  $poll(t) = \frac{1}{|Sel(P_t)|} \sum_{p \in Sel(P_t)} f(s(p))$  describing, in general form, an increase of the estimate  $poll(t)$  with an increasing number of high preference values  $s(p)$ . The simplest instantiation of such an aggregation is an *Averaging model* with all postings selected and no transformation of scores (i.e.,  $f$  being the identical function):  $poll(t) = \frac{1}{|P_t|} \sum_{p \in P_t} s(p)$ . Alternatively, we can apply a *Counting model* using thresholds on the sentiment scores:  $Sel(P_t) = \{p \in P_t | s(p) < thres_1 \vee s(p) > thres_2\}$  where  $thres_1$  and  $thres_2$  are thresholds used for discarding objective postings for which there is no clear sentiment assigned. Scores exceeding the thresholds can then be transformed into explicit binary “votes” to be counted. These votes are then averaged over as described above.

## Adjusting Aggregated Opinion Estimates

The assumption that blog articles fully reflect the opinion of the overall population is rather unrealistic. We want to address this issue and introduce model parameters to adjust the poll estimation function  $poll(t)$ . The first issue to account for are publishing delays: people can express their opinions in blogs at any point in time while phone interviews are carried out periodically. In order to address this issue, we introduce a *lag* factor that shifts  $poll(t)$  earlier or later in the timeline. Although blog writing technology has become more and more accessible to a broad range of people, users writing blogs are not necessarily representative for the whole population. To account for this difference we introduce an additive *bias* constant to our model. Finally, sentiment values computed as described in the previous subsection might require some re-scaling in order to reflect the actual “strength” of the opinions. We therefore introduce a constant multiplicative factor (*scale*) for the produced estimate. Considering all these factors we transform poll estimation for a time interval  $t$  in the following way:

$$AdjustedPoll(t, lag, bias, scale) = (poll(t+lag) + bias) \cdot scale$$

To account for general noise in the data we introduce a *smoothing* function over the estimates. We apply a simple moving average over past estimates:  $AdjustedPoll(t, k) = \frac{\sum_{j=0}^{k-1} poll(t-j)}{k}$  where  $k$  is the number of past time intervals considered for smoothing.

## Supervised Techniques to Improve Poll Estimations

In addition to the blog information a history of traditional opinion poll values might be available, resulting in a supervised scenario in which model parameters can be learned. We *optimize parameter values* introduced in the previous section by minimizing the average root mean squared error of  $poll(t, lag, bias, scale)$  compared to the values obtained from poll results of the given “training” history. In detail, we learn the best parameter values using the Nelder and Mead simplex approach (Nelder and Mead 1965). Specifically, the objective function is the following:

$$\operatorname{argmin}_{lag, bias, scale, k} \sqrt{\frac{1}{n} \sum_{t=1}^n (p(t, lag, bias, scale, k) - gt(t))^2}$$

where  $gt(t)$  is the traditional poll value for the  $t$ -th time interval. Furthermore, we apply TSA techniques in order to predict the continuation of a data series of opinion polls. In this paper, we use linear forecasting (Wei 2006) and polynomial regression (Cowpertwait and Metcalfe 2009) for predicting values  $pTSA(t)$  of the poll at time  $t$  which can, for instance, be linearly combined with the blog data-driven prediction  $poll(t)$ :  $poll_{TSA+blog}(t) = \lambda \cdot AdjustedPoll(t, lag, bias, scale, k) + (1 - \lambda) \cdot pTSA(t)$  where  $\lambda$  is a tuning parameter for controlling the influence of one or the other type of prediction.

## Experimental Evaluation

**Scenario and Data.** For computing opinion estimates about the two main rival candidates Obama and McCain we applied our methods to the *Blogs08* dataset (Macdonald, Ounis, and Soboroff 2010) which is composed of 28,488,766 blog postings from the 14th January 2008 to the 10th February 2009. Using a language classifier (Cavnar and Trenkle 1994) we estimated that about 65% of the postings are in English. Building an inverted index after stemming and removing stopwords from the English postings, we retrieved 670,855 matching the disjunctive query “obama OR mccain” which formed our working set.

As *ground truth* for opinion polls we used the data provided by Gallup, a professional service, tracking public opinion by means of telephone polls. This dataset provides results of interviews with approximately 1500 US national adults conducted over the time period from March until November 2008 for a total of 230 polls.

**Preprocessing and Setup.** We processed the set of relevant postings using state-of-the-art techniques for web page template removal (Kohlschütter, Fankhauser, and Nejd1 2010) in order to restrict our analysis to the actual content of the postings. We then used NLP tools<sup>1</sup> to split postings into a set of sentences and to POS-tag them. On the resulting output we did lookups in SentiWordnet to obtain positivity and negativity scores for each adjective in a sentence. For the ML-based sentiment score assignment, preprocessed sentences were classified using SVMlight (Joachims 1999) to obtain positive/negative scores.

In order to evaluate the effectiveness of the proposed models, we computed the Root Mean Squared Error (RMSE) between the estimation and the true poll value:

<sup>1</sup><http://gate.ac.uk/>

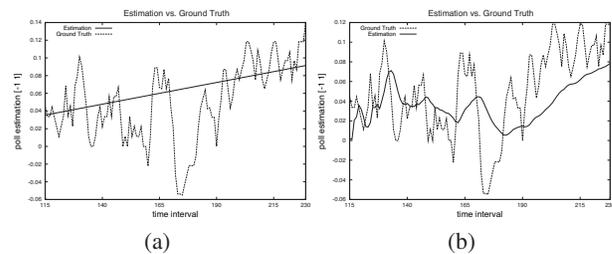


Figure 2: (a) Estimation of opinion polls as performed by a linear forecaster (LinFor) and (b) by the combination of a linear TSA model with our supervised Counting Model (LexCountLinFor).

$$RMSE(p, gt) = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - gt_i)^2}$$

where  $p_i$  is the estimate and  $gt_i$  is the poll value for the  $i$ -th time interval.

In the *unsupervised* setting, the task is to estimate the temporal development of opinions given only the blog data while for the *supervised* setting we assume that a history of past polls is given. Our experiments use the first 50% of the ground truth for learning model parameters and the remaining 50% for evaluating the models. We evaluate also the unsupervised models on the last 50% of the data in order to get comparable results.

**Evaluation Results.** We want to examine whether our sentiment aggregation methods are useful for predicting public opinion trends and whether supervised techniques as well as TSA methods can help at this task. We first measured the quality of the purely data-driven unsupervised models and aggregation techniques: the lexicon-based model using SentiWordNet to compute sentiment of postings and the Averaging aggregation model (**LexAvg**), the classification-based model using an SVM classifier trained on three opinionated datasets together with the Averaging aggregation model (**ClassifyAvg**), the lexicon-based model with the Counting aggregation model (**LexCount**), and the classification-based model together with the Counting aggregation model (**ClassifyCount**). For these techniques, we also test the performance of their supervised counterparts where parameters are learned on the initial 50% of the data.

Moreover, we compare against TSA methods that only exploit past available ground-truth to estimate future trend of opinions: a linear model based on forecasting techniques (i.e., Hunter’s simple exponential smoothing forecast model (Wei 2006)) (**LinFor**), a linear regression model to fit a deterministic trend to training data (Cowpertwait and Metcalfe 2009) (**LinReg**), and a polynomial regression model of degree 2 (Cowpertwait and Metcalfe 2009) (**QuadReg**). Finally, we compute the linear combination of our best supervised approach in terms of RMSE, **LexCount**, with the best TSA model, **LinFor**, and denote it **LexCountLinFor**.

In addition, we compare the proposed techniques with a very simple baseline just taking into account the plain number of retrieved postings for each candidate in a time interval. More specifically, we compute  $poll(t) = \frac{o+om}{o+m+om} - \frac{m+om}{o+m+om}$  where  $o$  indicates the number of postings in  $t$  about Obama,  $m$  about McCain, and  $om$  about both (**Count1**) and another variant where the postings containing both entities are not taken into account (**Count2**).

Table 1: RMSE values for the estimation models on the last 50% of the timeline. The initial 50% are used for training the model parameters of the supervised models.

Method	RMSE	
	unsupervised	supervised
Count1	0.1272	0.0653
Count2	0.3422	0.0632
LexAvg	0.0642	0.0556
ClassifyAvg	0.0619	0.0608
LexCount	<b>0.0572</b>	0.0483
ClassifyCount	0.1980	<b>0.0482</b>
LinFor	0.0397	
LinReg	0.0405	
QuadReg	0.0999	
LexCountLinFor	<b>0.0394</b> ( $\lambda = 0.2$ )	

Table 1 shows the RMSE values for all of the compared approaches<sup>2</sup>. Figures 1, 2a, and 2b show the detailed temporal developments for the best unsupervised, time series-based, and best overall approach (supervised data-driven learning linearly combined with TSA) respectively; the ground truth using traditional polls from Gallup is shown as dotted line.

All methods using sentiment-based blog analysis approximately capture trends, peaks, and switches in public opinions. Even purely data-driven unsupervised methods using aggregated sentiments extracted from the blogosphere can already estimate public opinion about electoral candidates quite well (RMSE = 0.0572 for **LexCount**). Figure 1 further illustrates the ability of unsupervised **LexCount** to match opinion changes. The supervised counterparts of sentiment-based methods that exploit the history of past opinion polls improve the quality of the up to 76% (RMSE = 0.0482 for **ClassifyCount**) compared to the unsupervised approach by tuning model parameters that take into account the intrinsic sample bias. The improvements over the unsupervised approaches are statistically significant (t-test  $p < 0.05$ ) in all the cases. Learned parameters are lag, taking into account time delays in estimations, smoothing to reduce noise into the underlying data, bias to tackle the problem of a non-representative sample, and scale to either amplify or reduce the estimation. The combination of TSA techniques with supervised models learned on the available data (**LexCountLinFor**) shows the best performance of all approaches resulting in overall statistical significant improvement of 39% compared to the best unsupervised approach (**LexCount**) and of 69% compared to the best approach based on posting volume (**Count1**) (see also Figure 2b for the temporal development). Simply counting postings published about a candidate over time results in a comparatively large estimation error (with RMSE = 0.1272 and 0.3422 for **Count1** and **Count2** respectively), and, thus, is not suited for the opinion estimation problem (see Table 1). The approaches based on sentiment classification and based on lexica show comparable performance. Forecasting methods, while respecting the general trend of the public opinion, do not provide an indication of peaks and switches (Figure 2a),

<sup>2</sup>Experiments show that the best thresholds for the Counting model are (0,0), that is, taking all postings into consideration.

in contrast to our data-driven models. On the other hand, the combination of TSA with our models results in a low RMSE and better captures changes of the public opinion (Figure 2b).

## Conclusion and Future Work

In this paper we presented an approach for estimating the development of public opinions over time by extracting and aggregating sentiments about politicians from the blogosphere. Our experimental study in the context of the US 2008 election campaign showed that purely data-driven, unsupervised approaches can already capture trends, peaks, and switches for public opinions in the political domain. We achieved further improvements by utilizing, in addition to blog data, a history of past opinion poll results for learning parameters that capture intrinsic differences between blogosphere and the “real world”. In our future work, we aim to enhance our approach taking into account various user characteristics such as gender, age, or location.

## References

- Cavnar, W. B., and Trenkle, J. M. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94*, 161–175.
- Cowperton, P., and Metcalfe, A. 2009. *Introductory Time Series with R*. Springer.
- Esuli, A., and Sebastiani, F. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6. Citeseer.
- Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Joachims, T. 1999. Making large-scale support vector machine learning practical. *Advances in kernel methods: support vector learning* 169–184.
- Kohlschütter, C.; Fankhauser, P.; and Nejd, W. 2010. Boilerplate detection using shallow text features. In *WSDM '10*, 441–450.
- Liu, Y.; Huang, X.; An, A.; and Yu, X. 2007. ARSA: a sentiment-aware model for predicting sales performance using blogs. In *SIGIR '07*, 607–614. ACM.
- Macdonald, C.; Santos, R. L.; Ounis, I.; and Soboroff, I. 2010. Blog track research at TREC. *SIGIR Forum* 44:58–75.
- Macdonald, C.; Ounis, I.; and Soboroff, I. 2010. Overview of the TREC 2009 Blog track. *TREC 2009*.
- Nelder, J., and Mead, R. 1965. A simplex method for function minimization. *The computer journal* 7(4):308.
- O’Connor, B.; Balasubramanian, R.; Routledge, B. R.; and Smith, N. A. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *ICWSM*.
- Ounis, I.; Macdonald, C.; and Soboroff, I. 2008. Overview of the TREC 2008 Blog Track. In *TREC 2008*.
- Pang, B., and Lee, L. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.
- Wei, W. 2006. *Time series analysis: univariate and multivariate methods*. Addison-Wesley.