Forecasting with Twitter Data

MARTA ARIAS, ARGIMIRO ARRATIA, RAMON XURIGUERA,
Universitat Politècnica de Catalunya

The dramatic rise in the use of social network platforms such as Facebook or Twitter has resulted in the availability of vast and growing user-contributed repositories of data. Exploiting this data by extracting useful information from it has become a great challenge in data mining and knowledge discovery. A recently popular way of extracting useful information from social network platforms is to build indicators, often in the form of a time series, of general public mood by means of sentiment analysis. Such indicators have been shown to correlate with a diverse variety of phenomena.

In this paper we follow this line of work and set out to assess, in a rigorous manner, whether a public sentiment indicator extracted from daily Twitter messages can indeed improve the forecasting of social, economic, or commercial indicators. To this end we have collected and processed a large amount of Twitter posts from March 2011 to the present date for two very different domains: *stock market* and *movie box office revenue*. For each of these domains, we build and evaluate forecasting models for several target time series both using and ignoring the Twitter-related data. If Twitter does help, then this should be reflected in the fact that the predictions of models that use Twitter-related data are better than the models that do not use this data. By systematically varying the models that we use and their parameters, together with other tuning factors such as lag or the way in which we build our Twitter sentiment index, we obtain a large dataset that allows us to test our hypothesis under different experimental conditions. Using a novel decision-tree-based technique that we call *summary tree* we are able to mine this large dataset and obtain automatically those configurations that lead to an improvement in the prediction power of our forecasting models. As a general result, we have seen that non-linear models do take advantage of Twitter data when forecasting trends in volatility indices, while linear ones fail systematically when forecasting any kind of financial time series. In the case of predicting box office revenue trend, it is support vector machines that make best use of Twitter data. In addition, we conduct statistical tests to determine the relation between our Twitter time series and the different target time series.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning; G.3 [**Probability and Statistics**]: Time series analysis; I.5.4 [**Pattern Recognition**]: Text Processing

General Terms: Algorithms, Performance, Experimentation

Additional Key Words and Phrases: Box office, Forecasting, Sentiment index, Stock market, Twitter

ACM Reference Format:
Arias, M., Arratia, A., and Xuriguera, R. 2012. Forecasting with Twitter Data. ACM Trans. Intell. Syst. Technol. V, N, Article A (April 2012), 25 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

Research supported by Spanish Government MICINN under grant TIN2011-27479-C04-03 and by SGR2009-1428 (LARCA). A. Arratia is additionally supported by grant SINGACOM MTM2007-64007.
Author's address: Universitat Politécnica de Catalunya, Departament de Llenguatges i Sistemes Informátics Campus Nord, Edif. Omega, Barcelona, Spain.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

The popularity of social networking platforms such as Twitter or Facebook has resulted in the creation of huge repositories of user-generated content to a large extent available through the internet. Exploiting this wealth of information in a timely manner has become of strategic importance to companies, health organizations and even government agencies. The scientific challenges of extracting useful information from this vast source of data are great due to its diversity and lack of formal structure.

In this work, we look at data from the popular microblogging site *Twitter*. In Twitter, users form social networks with other users that allow them to broadcast short messages of free text called *tweets*. Tweets can be about any topic, and it is totally up to the user what he or she wants to broadcast. They are publicly available through Twitter APIs; their availability has made them a very popular source of information for academic researchers, companies and other institutions.

By downloading huge numbers of tweets and using appropriate natural language and sentiment analysis techniques, it is now possible for example to have an approximate idea of what the general mood is at a given place and time, or for a particular topic of interest. This line of work has had a lot of success recently; researchers build a sentiment index from a Twitter collection of microblogs and correlate this temporal record with other target time series, for example: presidential polls [O'Connor et al. 2010], stock prices [Zhang et al. 2010; Wolfram 2010; Bollen et al. 2011], box office revenue [Mishne and Glance 2005], TV ratings [Wakamiya et al. 2011], or influenza rates [Lampos et al. 2010]. Common to many of these experiments is the fact that a Twitter sentiment index is used either as single predictor or in combination with a particular machine learning model. Results vary and conclusions abound, and confronted with this plethora of outcomes we asked ourselves: *Does Twitter really help?* In fact, we want to go further and want to know under which conditions Twitter does help.

To this end we have conducted a large set of experiments involving several machine learning models reinforced with a sentiment index built from Twitter data; we compare their performance when trained with and without the help of Twitter. If Twitter does contain useful information for the task at hand, then better prediction performance should be expected from the models that do use Twitter data. The strength of our paper lies in the fact that, contrary to much of the previous work where they test for simple correlations between Twitter and target signal or make use of a single model, we test three popular forecasting model families (linear models, support vector machines, and neural networks) and vary systematically their parameters, as well as other parameters related to the construction of the Twitter index and the way the experiments are set up. With this, we attempt to cover many experimental scenarios in order to identify the ones that lead to better predictions when Twitter data is available. In addition, and as an initial step, we have conducted statistical tests to determine non-linearity and causality relationships between the Twitter signal and the target. We have selected two different domains in which to test whether Twitter helps: the stock market and movies box office revenue. These have been selected for the following reasons: availability of the data, and the fact that they have been shown to have a positive correlation to Twitter or other internet signal.

In order to cope with the thousands of results obtained under the different experimental settings, we have developed a decision tree-based summarization method of this information which we call *summary tree*. A summary tree identifies those parameter settings under which the forecasting prediction with Twitter is superior to the forecasting prediction without Twitter.

The paper is structured as follows. Section 2 contains a description of work related to forecasting using data from the internet with main focus on Twitter. Section 3 describes everything related to data collection, data preprocessing, as well as the techniques used in the many parts of this work. Then, we show in Sections 4 and 5 the results obtained in our two application domains, namely, stock market and box office revenue. We finish with conclusions and future work in Section 6.

2. RELATED WORK

Several attempts at incorporating internet related data for making predictions have been done during the last few years. Table I shows a comparison of previous work in this direction. As can be seen from the table, some of the approaches do not consider any prediction model and only look at correlation between the additional and the target data. In this section we will describe the ones that are most relevant to this work.

In the realm of stock market forecasting using data from Twitter, one recent experiment that attracted considerable attention is by Bollen et al. in [Bollen et al. 2011] (included in the comparison table) in which the general mood of the messages published in Twitter is estimated and later used to predict the Dow Jones Industrial Average (DJIA). In this work, the general mood is estimated with two different lists of words. First, a general sentiment time series is created by computing the daily ratio between the amount of messages that contain positive words and the messages with negative ones. These positive and negative words are taken from the OpinionFinder[1] software. The other approach for estimating the general mood from Twitter messages is based on the extension of *Profile of Mood States - Bipolar* (GPOMS for short) rating scale. This extended list is not available to the general public, we have therefore not been able to incorporate these mood factors to our experiments. Once the general mood has been computed, they aim to predict the DJIA by providing these newly-created time series to a Self-Organizing Fuzzy Neural Network (SOFNN). In this work we generalize this setup to the most popular families of machine learning models and propose a framework to detect which of these models is the most accurate under different experimental settings.

Along the same line, Sebastian Wolfram [Wolfram 2010] also attempted to predict the price of some NASDAQ stock quotes by using Twitter as an additional source of information. His work differs in a variety of ways from ours, though. First, the features extracted from text are directly fed into the prediction model. That is, the intermediate step of analysing the messages' sentiment and creating a time series from it is omitted. Second, high frequency data from the stock market is used for the predictions. As stated in his work, historical data of this kind is not easily available, so predictions are only done for a two-week timespan. However, considering that consecutive observations are just one minute apart, this period is enough for testing the system. Regression is used to estimate the most immediate stock price. In addition, his work also incorporates a simulated trading engine to test how his system would perform in real life and how it would translate in terms of benefits.

The literature on box office prediction using messages from social networks is somewhat more limited in terms of forecasting. While both [Mishne and Glance 2005] and [Asur and Huberman 2010] take a similar approach in using sentiment analysis to explore the opinion of the general public in relation to specific movies, predictive power is not thoroughly explored. In [Mishne and Glance 2005], Nigam and Hurst only look at Pearson's r-correlation between some sentiment metrics derived from blog posts and the sales of 49 movies. Asur and Huberman [Asur and Huberman 2010] go a bit further and use linear regression to predict sales.

The principal aim of our work is to test whether using Twitter helps in making better predictions. What differentiates us mainly from previous work is the fact that we

[1] `http://code.google.com/p/opinionfinder/`

have tested this hypothesis under a wide variety of conditions, applying an extensive list of predictive models with varying parameter settings and testing for two different domains under a unified set of techniques.

3. DATA AND METHODS

3.1. Overview

This work involves the integration of many techniques. This section attempts to give an overview of all of these parts so that the reader can understand and place in context all of the methods covered in the subsections that follow. Figure 1 shows the transformation and handling of the data from source to result.

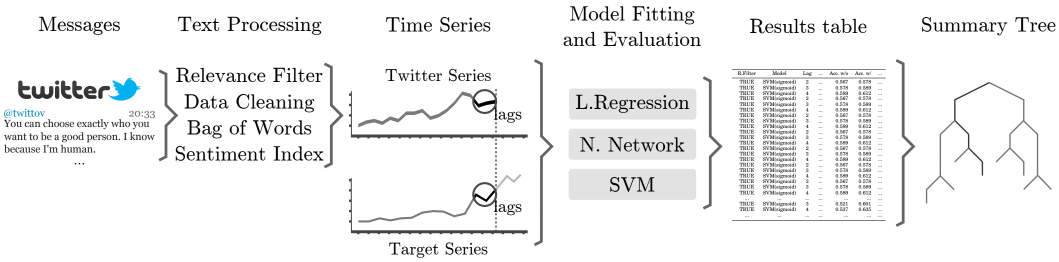

Fig. 1. Overview of data collection, preprocessing, forecasting and final analysis processes.

Our framework for the study of forecasting with Twitter data proceeds in three stages. The first one deals about collecting Twitter data, cleaning and preprocessing in order to create a sentiment index (cf. Sections 3.2–3.4). Once we obtain the sentiment index, we apply statistical tests to determine the relation between the time series corresponding to the sentiment index and the target time series (cf. Section 3.5). Then, we proceed to apply our forecasting models to predict the target time series both using the sentiment index and not using it, obtaining a table of results which tells us, for each experimental scenario given by a particular assignment of values to parameters, the observed accuracy with the index and without it (cf. Sections 3.6 and 3.7). Finally, we use our summary trees (cf. Section 3.7) to explore the answer to the main question posed in this paper, namely, whether Twitter helps.

3.2. Twitter Data

Twitter is an online microblogging platform that allows its users to build social networks. Its core functionality is to share messages with the members of one's own network, known as *followers*. In the Twitter domain, these messages are known as *tweets* and are limited to a maximum of 140 characters by design. Figure 2 shows an example of a tweet; it has some parts worth noting:

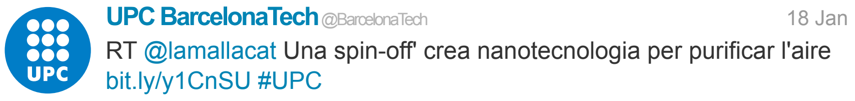
UPC BarcelonaTech @BarcelonaTech 18 Jan
RT @lamallacat Una spin-off' crea nanotecnologia per purificar l'aire bit.ly/y1CnSU #UPC

Fig. 2. Example of a tweet.

Table I. Table comparison of related work on forecasting using available data from the Internet.

Ref.	Event	Models	Corpus	Conclusion
[Wolfram 2010]	NASDAQ stocks	SVM	Edinburgh Corpus, English, Relevant to stocks	Works with high freq. data. No sentiment analysis, but direct count of frequency of words
[Zhang et al. 2010]	DJIA, S&P500, NASDAQ, VIX	n/a	English, with mood keywords	Finds correlations of tweet's emotions (hope, fear, worry) and the direction of the DJIA stock index.
[Bollen et al. 2011]	DJIA	SOFNN	$\sim$10M tweets, Stock market prices	An index of the calmness of the public is predictive of the DJIA and predictions can be significantly improved using a SOFNN.
[Mishne and Glance 2005]	Movie sales	n/a	Blog posts with links to IMDB, IMDB sales data	Considering the sentiment of blog posts improves the correlation between references to movies and their financial success.
[Asur and Huberman 2010]	Movie sales	Linear regression	$\sim$2.9M tweets for 24 movies	The model built with the tweet rate time series outperforms the baseline that uses the Hollywood Stock Exchange (HSX).
[O'Connor et al. 2010]	U.S. polls	n/a	10^9 tweets (omitting non-English), Public opinion polls	The evolution of Twitter sentiment correlates to periodical public polls on the presidential election and on the presidential job approval.
[Tumasjan et al. 2010]	German 2009 election	Logistic Regression	100K tweets	Additional information is not provided to predictive models. Only a comparison of the share of voice and the election results.
[Gruhl et al. 2005]	Book sales	Custom *Spikes* predictor	Blog posts, Amazon sales rank	Correlation detected between the number of blogs refering to a book and its sale spikes.
[Wakamiya et al. 2011]	TV Ratings	n/a	Japanese tweets showing hashtags related to TV programs	No predictions or correlations are made; only identifes tweets that talk about TV programs.
[Ritterman et al. 2009]	Swine Flu Pandemic	SVM	Edinburgh Corpus, Newspaper articles	Adding features concerning historical context of a feature has a beneficial impact on forecast accuracy.
[Lampos et al. 2010]	UK influenza rates	Sparse linear regression	200K Twitter posts from 49 cities in the UK	Statistical correlation established between predictions and ground truth of flu rates.
[Culotta 2010]	US influenza rates	Logistic Regression	Over 500M tweets	Similar to [Lampos et al. 2010].

- Other people can be *mentioned* or *replied to* by using the `@` symbol followed by their user name. User names are alphanumeric strings of up to 15 characters. Underscores are allowed as well.
- Tweets beginning with the expression `RT @[\w_]{1,15}` are called *retweets* and are a handy way to share information with the people in your network.
- Words within a message preceded by the # symbol are known as *hashtags* and are mostly used to assign messages to topics or to mark keywords.
- Finally, a tweet can also contain URLs. While this is not Twitter-specific, it is important to note that links are very common and are to be expected.

Retrieving Tweets. Even though working with Twitter data is becoming very common, one major problem is the lack of standard datasets. In April 2010 Twitter updated the API terms of service introducing a rule that does not allow third parties to redistribute Twitter Content without the company's prior written approval [Twitter 2010b; Twitter 2010a]. Therefore, attempts to release Twitter corpora, like the Edinburgh Corpus presented in [Petrović et al. 2010], have failed. These restrictions make it very hard to reproduce previous results; thus, we had no choice but to collect our own data using the APIs permitted by Twitter, which work under restricted terms of use imposing bounds on the number of requests per hour that can be made, volume of tweets that can be retrieved, and other limitations. After evaluating all these APIs we chose to work with the Twitter Streaming API, which is intended for developers with data intensive needs and it works by establishing a single HTTP long-lived connection that is kept alive indefinitely and over which new tweets are sent as they are being posted; it also has a filtering method which is very convenient for the task we are trying to accomplish.

The retrieval of tweets (or *listening*) began on 22 March 2011.

3.3. Preprocessing Twitter Data

We applied standard data cleaning and preprocessing techniques for preparing the Twitter data to build the sentiment index. These techniques include lower-case conversion, stop-word removal, duplicate removal (mostly corresponding to retweets), and language detection. We also have experimented with an automatic relevance-filtering method based on Latent Dirichlet Allocation (LDA). In what follows we explain some of these techniques in more detail.

Language detection. Twitter's user interface is currently translated to 22 languages [Twitter 2012] and the default language is English. Therefore, it does make sense to do some sort of language detection. We use the Guess Language[2] library, to associate a language to each tweet at the time of retrieval. Internally, this tool applies some heuristics and looks at the frequencies of *trigrams* for each of the considered languages. The brevity of the tweets means that this method may fail to detect the correct language much more often than what it would for more extensive texts. This problem worsens if we take into account that people tend to use some English words in their native languages.

Handling negation. Negation can play an important role in the task of sentiment classification. Consider the sentences *I think it was good* and *I think it was not good*. While they only differ in one word and would score highly in most of similarity measures, their sentiment polarities are completely opposite. We have attempted a very simple form of negation handling for English texts by tagging words between common polarity shifters such as *not*, *don't* or *haven't*. For instance, the sentence from the previous example would become *I think it was not NOT_good*. With this transformation, a word

[2] http://code.google.com/p/guess-language/

and its negated counterpart are considered to be different words, significantly increasing the size of the vocabulary. This translates to a larger set of features thus a larger dataset is preferable. Moreover, this technique only covers a small subset of negations where there is a valence shifter involved. This is the same approach as the one used in [Das and Chen 2001], also described in [Pang and Lee 2008], Section 4.2.5, although more complex techniques that take into account part of speech tags are also possible (see e.g., [Polanyi and Zaenen 2006] and [Potts 2010]).

Relevance filter. One of the problems found by skimming through the collected data is that there is a considerable amount of tweets that, while containing one or more of the filtering keywords, are not relevant to the task we are trying to accomplish. This is usually caused by homonyms, polysemous words, proper names or words that are part of an idiom. We attempt to alleviate this problem by using Latent Dirichlet Allocation (LDA), a generative probabilistic model mostly used for topic modelling [Blei et al. 2003], built upon Latent Semantic Indexing (LSI) and Probabilistic LSI. During training, LDA is provided with relevant tweets from the tweet collection and the algorithm generates a latent description of these tweets, thus creating a profile for what we consider relevant. Future tweets will only be considered if they conform to this profile, thus filtering out those that the LDA model thinks are irrelevant. We employ an implementation of the variational Bayes algorithm for online LDA introduced by Hoffman et al. [2010] and which is available at Hoffman's webpage[3].

We have created three different datasets with which we will train three different LDA-based filtering models. The first dataset consists of 16000 tweets containing dollar-tagged symbols as relevant to the stock market. The second one has been created by retrieving tweets from users that usually post stock-related messages. This second dataset contains 10000 random tweets from our collection plus a total of 9978 tweets from a list of twenty accounts including *FinancialTimes*, *BBCBusiness*, as well as some investors and bloggers like *Dan Tanner* or *Jim Cramer*. Finally, we have created a third dataset by just taking 300000 tweets from our collection, mostly containing company names.

To evaluate the performance of the three trained models, we have created an independent test set of 4000 tweets, 2000 tweets from the aforementioned stock-related accounts labelled as relevant and another 2000 supposedly irrelevant tweets from Twitter accounts we are fairly certain that do not focus on the stock exchange. These include accounts from musicians, comedians or personal bloggers among others. It should be noted, though, that these labels have been set automatically, and so large errors may be expected.

Using the three datasets described above, we have trained three models setting the number of latent topics to 2. Results of the performance of these three models are shown in Table II.

Table II. Performance of LDA filters. Different filters have been trained using the training datasets described above. Performance is measured over an independent test dataset of 4000 Tweets.

Dataset	Accuracy	Precision	Recall	F-measure
Dollar-tagged	54.88%	53.86%	**67.98%**	60.10%
Stock-related accounts	64.21%	64.35%	63.68%	64.01%
300K from collection	**76.16%**	**83.31%**	65.43%	**73.29%**

[3] http://www.cs.princeton.edu/~mdhoffma

3.4. Sentiment Classifier and Sentiment Index

The goal of a textual sentiment classifier is to determine whether a text contains positive or negative impressions on a given subject. After a preprocessing step, where at this stage the noisy terms from the tweets are removed (cf. Section 3.3), the next problem that must be addressed is the development of a corpus from which to train a sentiment classifier.

As a first step for constructing such a corpus, we implemented in this project a recent labelling idea which consists on automatically tagging a tweet as positive if it contains one of the *smileys* :-), :-D, or negative if it contains :-((see e.g. [Go et al. 2009; Bifet and Frank 2010]). In fact, the actual list of tweets we consider is somewhat wider, using the regular expressions `[:8=][- ]?[)D]` and `[:8=][- ]?[(]` for positive and negative tweets respectively. Some care must be taken in this primary sentiment division, since as noted in [Go et al. 2009; Bifet and Frank 2010], there are much more tweets containing positive smileys than negative ones.

Several attempts at multi-class sentiment analysis have been made. For instance, in [Ahkter and Soria 2010] Facebook messages are classified into *Happy*, *Unhappy*, *Sceptical* and *Playful*. Nevertheless, we only focus on the binary classification problem.

Multiple datasets have been extracted from our tweet collection for training the sentiment classifier. Each dataset has a balanced number of positive and negative instances and thus the total amount of tweets is limited by the number of negatives in the collection. Retweets have not been included in this datasets.

Table III. Properties and size of training datasets

Tweet language	Smiley Location	Training Instances
English (en)	End of tweet	380000
English	Anywhere	600000
Multi-language (ml)	End of tweet	1300000
Multi-language	Anywhere	1800000

We have trained several sentiment classifiers using Multinomial Naïve Bayes with varying preprocessing steps and multiple sets of feature words to represent the documents for each of datasets of Table III. Three feature lists have been considered: a general one with frequent words from the aforementioned training datasets and two additional lists with frequent words from two collections of tweets related to the stock market and films. To evaluate the classifiers generated, we have collected two additional independent datasets (English and multi-language), containing 50000 and 200000 tweets respectively. Only the best scoring classifiers are listed in Table IV and are considered throughout the rest of this paper.

Table IV. Best scoring sentiment classifiers. Accuracies reported are over independent test datasets.

Label	Feature List	Tweet Language	Smiley Loc.	Hashtag	Vect.Values	Accuracy
C-En	General	English	End	Replace	Frequency	76.5 %
C-Ml	General	Multi	Any	Keep	Frequency	79.5 %
C-Stk	Stock	Multi	Any	Keep	Frequency	76.1 %
C-Flm	Films	Multi	Any	Replace	Presence	76.0 %

Bear in mind that the test datasets are automatically labelled using the smiley approach and large errors are expected. The sentiment classifier that we use has been built using standard methodology, and our results are comparable to most state-of-the-art methods, some publicly available through APIs as, for example, in [Go et al. 2009].

Sentiment Index. Using the top-scoring sentiment classifiers we obtain a collection of daily sentiment indices that will represent the evolution of the general mood towards a specific item, expressed in terms of one or more filtered words. Each sentiment index is represented as a time series where every value corresponds to the daily percentage of positive tweets over the total number of messages that were posted on a given date. It should be noted that, in contrast to the training of the sentiment classifier, retweets are not removed during the sentiment index generation. As a result, tweets with a high amplification have a greater impact on the final value of the index. In addition to the sentiment index, we consider the time series given by the daily tweet volume as an alternative indicator.

Throughout the paper, we refer to these additional Twitter-derived series as the *Twitter series*, which can either be a sentiment index (if sentiment analysis has been performed), or tweet volume index.

3.5. Model Adequacy

Modeling time series involves to a certain extent subjective judgement; nonetheless one can (and should) draw some general guidelines through statistical testing. We attempt to be mathematically rigorous and, hence, in order to have some certainty of the adequacy of Twitter as part of a forecasting model for our data sets, we run some widely accepted tests to assess, first, for a nonlinear relationship among the target time series and a series build from Twitter data, which can either be a sentiment index (as described in Section 3.4) or a simple count of tweets (i.e., volume); and second, test for causality from and to the Twitter series and the target time series, at different lags. We briefly comment in thist section on the tests we use for the assessment of our models.

Neglected nonlinearity. A multivariate test of nonlinearity to ascertain if two time series are nonlinearly related can be achieved with the neural network test for neglected nonlinearity developed by White [White 1989; Lee et al. 1993]. The basic idea is to perform a test of the hypothesis that a given neural network defines a perfect mapping between its input and output and that all the errors are due to randomness. For our experiments we use the Teräsvirta linearity test, presented in [Teräsvirta et al. 1993] and based on White's neural network test for neglected nonlinearity. An implementation of this algorithm is available in the `tseries` R library.

Granger Causality. We would also want to assess the possibility of causation (and not just correlation) of one random variable X towards another random variable Y. In our case X and Y being the time series under study. The basic idea of Granger causality [Granger 1969] is that *X causes Y, if Y can be better predicted using the histories of both X and Y than it can by using the history of Y alone*. Formally one considers a bivariate linear autoregressive model on X and Y, making Y dependent on the history of X and Y, together with a linear autoregressive model on Y, and then test for the null hypothesis of "X does not cause Y", which amounts to a test that all coefficients accompanying the lagged observations of X in the bivariate linear autoregressive model are zero. Then, assuming a normal distribution of the data, we can evaluate the null hypothesis through an F-test. This augmented vector autoregressive model for testing Granger causality is due to Toda and Yamamoto [1995], and has the advantage of working well with possibly non-stationary series. (An accepted characteristic of financial time series is that they are at best weakly stationary [Tsay 2010].) The test is implemented in R as the function `grangertest`, as part of the package `lmtest` for testing linear regression models.

Now, there are two major drawbacks on applying this parametric Granger test. One is that it assumes a linear dependence on variables X and Y; and the other is that it assumes data are normally distributed. The first limitation goes contrary to our presumed nonlinear relationship between the Twitter time series (whichever we build), the machine models we want to pair it up with, and the target time series; in particular any financial time series. The second limitation poses a strong assumption on the distribution of the data, which is seldom the case for stocks. Thus we shall alternatively apply a non–linear and nonparametric test for Granger causality hypothesis presented in [Diks and Panchenko 2006], implemented in C by the authors and publicly available.

3.6. Forecasting Models and Evaluation

As has been stated in the Introduction our primary goal is to study the effect of using Twitter data when doing predictions in different domains and jointly or not with various machine models popular among the Machine Learning community. Therefore, after processing the tweets as it has been explained above, we are left with two different time series: a target time series that we are attempting to predict and the Twitter series (volume or sentiment). The predicted values for the target time series are obtained by training a machine learning model and providing them with past observations (*lags*) of both the target and Twitter time series. Our predictions are limited to guessing the target times series' immediate direction, that is, whether the next future value will be higher or lower than the last observation. This is a binary classification task, so the models considered in the experiments have been chosen accordingly. The models that we have considered are among the most popular and effective in Machine Learning. These are: Linear Regression; Neural Networks (feedforward); and Support Vector Machines (with polynomial, radial or sigmoid kernel). Detailed descriptions of these models can be found in standard textbooks such as [Mitchell 1997; Hastie et al. 2003].

Time series forecasting evaluation is typically done by holding out an independent data set from the training data. However, since the amount of available data in our collection is rather limited in terms of the number of daily observations, we have taken a *prequential approach* [Gama et al. 2009; Bifet et al. 2010] for evaluating the experiments. This means that, for each prediction, a model is fitted with all the available past data. Once the actual value is known, it is included in the training set so it can be used for the next prediction. After repeating this process for all the available observations, we get a contingency table of hits and misses like the one below.

Table V. Contingency table.

Actual	Predicted		
	Up	Down	
Up	m_{11}	m_{12}	m_{10}
Down	m_{21}	m_{22}	m_{20}
	m_{01}	m_{02}	m

In the table, m is the total number of predicted values, m_{11} is the number of correct predictions (or hits) for the upward movement, m_{12} is the number of failed predictions (or misses) for the upward movement, and so on. The performance of each of the tested models is then measured with three different metrics:

Accuracy. Computed as a simple percentage of correct predictions: $(m_{11} + m_{22})/m$.

Cohen's Kappa (cf. [Cohen 1960]). This measure takes into account the probability of random agreement between the predicted and the actual observed values and it is computed as $\dfrac{P(a) - P(e)}{1 - P(e)}$, where $P(a) = (m_{11} + m_{22})/m$ is the observed agreement and $P(e) = (m_{10}m_{01} + m_{20}m_{02})/m^2$ is the probability of random agreement, that is, the probability that the actual and the predicted coincide assuming independence between predictions and actual values. Thus, $\kappa = 1$ when the predicted and the actual values completely agree.

Directional Measure (cf. [Tsay 2010]). This is computed out from the contingency table as $\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{(m_{ij} - m_{i0}m_{0j}/m)^2}{m_{i0}m_{0j}/m}$. The χ^2 behaves like a chi-squared distribution with 1 degree of freedom, and we can use this information to compute the quantile with respect to a given significance level. Similar to Cohen's Kappa, large values of χ^2 tell us that the model outperforms the chance of random agreement.

3.7. Experimental Setup and Summary Trees

In our experiments, we have tried all possible combinations of a large number of parameters. We have grouped these parameters into three categories reflecting the main components of our forecasting experiments.

(1) *Target time series*

Predicted symbol. For the stock market application, this parameter refers to the ticker symbol of the target company. For the movie box office application, it refers to the title of the film, which we list in Table XII.

Predicted Series. This parameter establishes the target time series that we are attempting to predict. In the case of stocks this can be set to *Returns* or *Volatility*. For implied volatility indices, we only consider the volatility, which is given as their closing value. For films it is just *Gross Revenue*.

Lag value. The number of observations that the time series is shifted back in time. It should be noted that the higher the lag, the fewer days that are available for training.

(2) *Twitter index*

Twitter Dataset. Defines from which of the datasets defined in Section 4.1 should the Twitter time series be extracted.

Twitter Series. Sets what kind of transformation should be applied to Twitter data in order to create a time series from it. Four possible values are defined: daily tweet volume time series, the sentiment index, the sentiment computed using OpinionFinder and a fourth possibility combining values from the volume and the computed sentiment index.

Relevance Filter. Whether Latent Dirichlet Allocation (LDA) has been applied to filter out non-relevant messages before creating the sentiment index series to add to the predictions. This filtering approach has only been applied to stock market Twitter data, where we have detected much more noise as opposed to the box office domain.

Sentiment Classifier. This parameter describes the classifier used to perform the sentiment predictions depending on the dataset that was used to train them. For stocks, we employ the classifiers *C-En*, *C-Ml* and *C-Stk*, and for films we use *C-En*, *C-Ml* and *C-Flm* (cf. Section 3.4). In both cases, we are considering the three best-scoring classifiers.

(3) *Forecasting model*

Model Family. Refers to one of the three models listed in Section 3.6, namely, support vector machines, with its different kernel flavors such as polynomial, radial and sigmoid; different-degree neural networks or linear models.

All the different parameter combinations result in a battery of tens of thousands of different experiments, each of them telling us the accuracy (and other performance metrics, see Section 3.6) using Twitter or not. From this table, we filter out those experiments that do not exceed the threshold of 50% of accuracy when using Twitter. This should exclude experiments that have very poor performance and therefore would only add noise to the result.

Attempting to draw any conclusions by just looking at such a long list of data is overwhelming and hence we need a succinct way of extracting the important information, performing the most relevant mixture of parameters. We have tackled this problem by automatically generating a decision tree that will tell us, in general, which of the different parameter combinations lead to an increase of predictive power. We call such a tree a *summary tree*.

In a summary tree, each of its intermediate nodes has a parameter assigned to it. Different branches spring from these nodes depending on the value of the parameter assigned to the given node. The leaves of the tree are tagged with the result (improves/-does not improve) of the given branch. Leaves also contain information on how many of the instances in that branch behave in the same way. In our experiments, a model is considered to have improved when adding the time series from Twitter results in an increase in prediction accuracy of 5% or more. The tree is generated using `REPTree`, a decision tree learner available in Weka[4] that builds trees by greedily selecting, for each node, which of the parameters will result in a higher information gain. One of the characteristics of the `REPTree` algorithm is that the height of the tree can be limited in advance. We have also run the experiments with Weka's implementation of C4.5 and results are consistent. Summary trees contribute to our analysis in that they help identifying sets of parameters – defined by the branches – that lead to consistently positive (or consistently negative) results. Even if the branches are over-specific, they can serve as a guide for deciding in what direction should we attempt to derive more general conclusions.

4. STOCK MARKET APPLICATION

4.1. Stock-related Twitter Data

For our experiments into forecasting the stock market with Twitter data, the following technology companies were selected: Apple (AAPL), Google (GOOG), Yahoo! (YHOO), Microsoft (MSFT). Both the company name and ticker symbols in parentheses were tracked in Twitter, covering a time span of eight months from 20 March to 20 November 2011. This translates to roughly 170 working days, after discarding weekends and U.S. federal holidays such as Memorial Day, Independence Day, and others. The focus on technology companies is due to the availability of a higher volume of user generated messages than, for instance, companies from the energy or healthcare sectors. The reason not to track more companies was mainly the rate limit mentioned in Section 3.2.

Apart from using all messages to train the predictors, we also split them into different datasets depending on the company they are related to. The number of available tweets in each dataset is shown in Table VI. Some stock market indices such as Standard&Poor's S&P100 and S&P500 are obtained by combining the prices of large corporations and weighting them based on their market capitalizations (share price × number of shares). The companies listed above are components of these S&P's indices

[4] `http://www.cs.waikato.ac.nz/ml/weka/`

Table VI. Available tweets per company.

Dataset	Multi-language	English
Yahoo!	2881410	2048301
Google	2353057	1076108
Apple	1588157	1204255
Microsoft	29790	21262
All previous four	6852414	4349926

and, in particular, both Apple and Microsoft are in the top ten market capitalization companies. Therefore, we believe that the combination of the tweets related to these companies can be of good use to predict these indices as well.

4.2. Stock Market Data

The following companies and market indices are targeted: Apple (AAPL), Google (GOOG), Yahoo! (YHOO), Microsoft (MSFT), S&P100 (OEX), S&P100's implied volatility (VXO), S&P500 (GSPC) and S&P500's implied volatility (VIX). We also consider as target the historic volatility of the stock price series of each of the companies. There are currently a wide variety of websites that offer daily stock market data for download. The historical price series for the above mentioned companies were retrieved from Yahoo! Finance[5]. Among the multiple daily values offered by this service, we are particularly interested on the Adjusted Close which corresponds to the closing price once it has been updated to include any dividend payment and corporate actions, such as splits of the value, occurred at any time prior to the next day's open.

Furthermore, instead of directly working with adjusted closes we focus on the price returns, or benefit. Returns are computed using the equation $R_t = (P_t - P_{t-1})/P_{t-1}$, where P_t is the price at time t. Assuming that the returns come from a log normal distribution (cf. [Tsay 2010]), then their logarithm is normally distributed. Thus, using logarithmic returns can be more convenient when working with statistical methods that assume normality. Moreover, for small changes, returns are approximately equal to their logarithmic counterparts, which are computed as $r_t = \ln(1 + R_t) = \ln(P_t/P_{t-1})$.

Historic daily volatility for a specific company can be estimated with the square of the returns of the m past days for that company. In our experiments a 30-day exponentially-weighted moving average (EWMA) is used. The main idea behind EWMA is that the square of returns of the last few days should have a greater impact on the volatility than the square of returns of last month. This is achieved by exponentially decreasing the weight of the square of returns as we get further in the past. Thus, historic volatility at time t_n is given by $V_n = (1 - \lambda)\displaystyle\sum_{i=1}^{m} \lambda^{i-1} R^2_{n-i}$.

One of the advantages of using EWMA in favour of other options is that it can be computed recursively with just the previous days' volatility and squared return: $V_n = \lambda V_{n-1} + (1 - \lambda)R^2_{n-i}$, where λ is usually set to 0.94, according to J. P. Morgan's Risk Metrics^{TM}. Implied volatilities for stock indices such as S&P100 (OEX) or S&P500 (GSPC) are much more complex and are computed with a pricing model such as Black&Scholes [Tsay 2010]. Implied volatilities are available from Yahoo! Finance as separate indices. S&P100 and S&P500 have implied volatilities VXO and VIX, respectively.

Finally, the standard z–score, $z = (x - \mu)/\sigma$ is employed for transforming the time series to have zero mean and a standard deviation of one. This kind of normaliza-

[5] http://finance.yahoo.com

tion is important when various attributes, such as tweet volume and log returns are expressed in different units, since it will bring all the series to the same scale.

4.3. Summary Tree

We have applied our summary tree technique to summarize a table of approximately 39000 different experiments, which we partly show in Table VII.

Table VII. Partial view of the table containing the full set of experiments.

Target	Dataset	R.Filter	Model	Lag	...	Acc. w/o	Acc. w/	...
GOOG	GOOG	TRUE	SVM(sigmoid)	2	...	0.567	0.578	...
GOOG	GOOG	TRUE	SVM(sigmoid)	3	...	0.578	0.589	...
GOOG	GOOG	TRUE	SVM(sigmoid)	4	...	0.589	0.612	...
...	...	...	...	...	...	...	...	...
VIX	YHOO	TRUE	SVM(sigmoid)	3	...	0.521	0.601	...
VIX	YHOO	TRUE	SVM(sigmoid)	4	...	0.537	0.635	...
...	...	...	...	...	...	...	...	...

A section of the resulting tree is shown in Figure 3. For the sake of readability, we have pruned the tree to only show those paths that lead to a clear win or loss in terms of number of experiments improving vs. not improving. The complete tree is reproduced in Appendix A in a more space-efficient format.

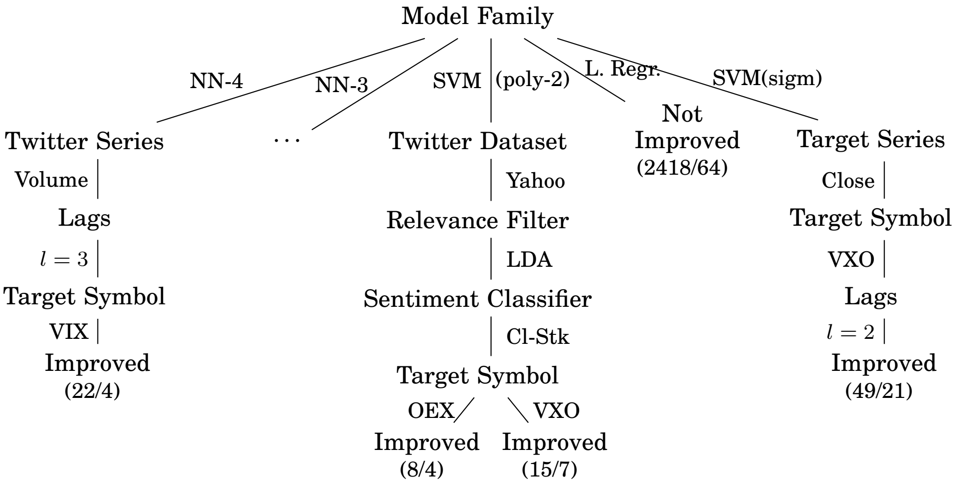

Fig. 3. Partial view of the summary tree for the stock market application.

What stands out most in the summary tree is the failure of linear models which only improve 2.5 % of the time. Furthermore, the fact that the tree did not branch out from this subnode means that no further filtering of this data would have resulted in a positive result.

Taking the parameters defined by the leftmost branch, we have analysed how the same set of parameters perform grouping by the three different model families we consider (linear models, support vector machines, and neural networks). The results table contains 399 experiments for predicting the VIX index using Tweet Volume and

$lags = 3$. In general, 219 out of these 399 experiments, that is 54.88%, yield successful results, but by separately looking at the different model families, some more information on the real capabilities of each family can be inferred. This is shown in Table VIII and, as can be seen, a majority of neural networks have successful results for this specific set of parameters.

Table VIII. Success rates by model family for predicting the VIX index using Tweet Volume and $lags = 3$.

Model family	Successful	Unsuccessful	Success rate
Linear Regression	16	24	40.00%
Neural Networks	100	35	74.07%
Support Vector Machines	103	121	45.98%

Analogously, if we take the rightmost branch of the tree, that is, predicting the VXO index with $lags = 2$, we obtain a general success rate of 43.24%. Comparing the different model families none of them achieved a success rate over 50 %. If we focus this time on SVMs and look at the different kernel types separately, we see that the success rate for the sigmoid kernel is far superior to the other kernels. Results are listed in table IX. The use of the model family or kernel types just serves as an example. This same analysis could be done by generalising any of the other parameters.

Table IX. Success rates of SVM by kernel type for predicting the VXO index when $lags = 2$.

Kernel Type	Successful	Unsuccessful	Success rate
Polynomial Kernels	104	192	35.13%
Radial	5	70	0.07%
Sigmoid	68	7	90.67%

4.4. Case Study: S&P 100 Implied Volatility (VXO)

From the many experiments that have been performed, we have selected for presentation and analysis the results for predicting the VXO ticker using the Yahoo dataset after applying LDA-filtering to remove non-relevant tweets. The *C-En* classifier (cf. Section 3.4) is used to build the sentiment index.

The VXO and tweet sentiment index time series for the period between 22 March and 18 November 2011 are depicted in Figure 4. It is difficult to see a clear correlation from the chart since sentiment data appears to be very irregular. Thus, we perform next the statistical tests introduced in Section 3.5 for a more in-depth study of the relation between the time series.

Nonlinearity. Teräsvirta's neural network test for neglected non-linearity is inconclusive; the p-values obtained for a 95% confidence interval are 0.52 for *VXO → Sentiment Index* and 0.60 for *Sentiment Index → VXO*.

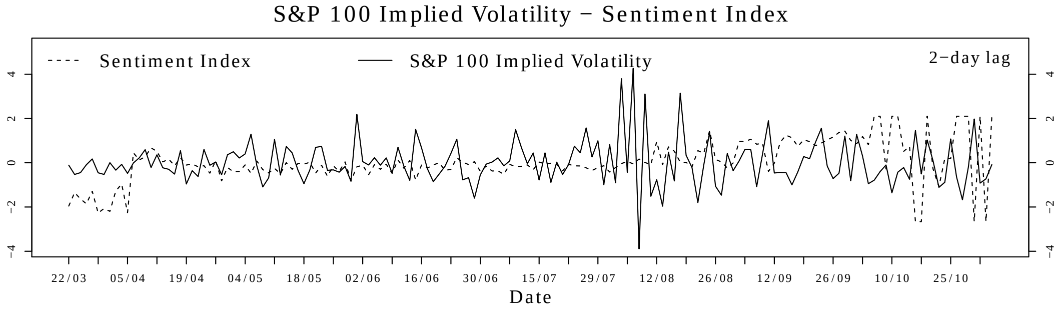

Fig. 4. S&P 100 implied volatility versus daily sentiment index.

Table X.

	Granger Causality		Non-parametric Causality	
Lag	VXO → Sent.	Sent. → VXO	VXO → Sent.	Sent. → VXO
1	0.7058	0.7931	0.8137	0.7385
2	0.5540	0.8220	0.2959	0.9022
3	0.6437	0.8842	0.4890	0.8522
4	0.5018	0.5972	0.2499	0.7800
5	0.6707	0.5901	0.1907	0.8044

Causality. Table X reproduce the p-values for the causality tests between target VXO and Twitter series. The tests are done in both directions, to and from the Twitter series, and by shifting the values from 1 to 5 days. In the tables, arrows indicate the direction of the causality and all p-values lower than 0.1 are shown in bold.

As can be seen from Table X, none of the p-values allow us to conclude the existence of causality between the two time series. We go on to evaluate the forecasting models.

Model fitting and evaluation. Table XI contains our results obtained by adding the daily sentiment index information for the prediction S&P 100 implied volatility (VXO). Even though the statistical tests do not suggest a causality relation between the series we can see that for lags 2 and 4 there is an improvement of the predictions when using the Twitter sentiment index.

Table XI. Effect of adding past values of daily sentiment index for prediction of the S&P100s implied volatility (VXO). SVM with a sigmoid kernel. The table shows a comparison of accuracies (Acc.), Cohen's Kappa statistics (Kappa) and Directional Measure p-values (DM) depending on whether Twitter information was used (w/) or not (w/o) for the predictions.

Lag	Acc. w/o	Acc. w/	Kappa w/o	Kappa w/	DM w/o	DM w/
1	0.6346	0.5576	0.2658	0.1065	0.0009	0.1786
2	0.5961	0.6794	0.1885	0.3581	0.0184	0.0000
3	0.6666	0.5833	0.3315	0.1672	0.0000	0.0364
4	0.6168	0.6558	0.2322	0.3140	0.0039	0.0001
5	0.6493	0.6168	0.2987	0.2353	0.0002	0.0033

5. BOX OFFICE APPLICATION

5.1. Film-related Twitter Data

The task of collecting tweets in the movie sales domain is a bit more complex. In contrast to the stock market, films have a short life span and are only mentioned for a limited time. Thus, the selection of titles that were tracked evolved over time.

A list of upcoming releases is maintained at *IMDB*[6]. Using this list, our tracking system started retrieving messages for upcoming movies two weeks before the release date and it kept collecting data for a minimum of four weeks. Movies still in the top ten box office ranking after two weeks from release were still tracked. Due to Twitter API restrictions the number of movies being tracked simultaneously was limited to 50.

The box office dataset that we collected spans from late June to August 2011 and consists of more than 100 million tweets from a total of 121 different films. However, only a small fraction of this data is used in the experiments presented in this section. Different datasets of tweets for eight of the most popular movies were created and are listed in Table XII.

Table XII. Available data from release until 24 August 2011

Abbrv.	Film	Tweets	Days	Tweets/Day
hp	Harry Potter and the Deathly Hallows: Part 2	1323779	41	32287
ca	Captain America: The First Avenger	572064	34	16825
ra	Rise of the Planet of the Apes	258760	20	12938
s8	Super 8	491017	75	6546
c2	Cars 2	284155	50	5683
sf	The Smurfs	144847	27	5364
cw	Cowboys & Aliens	136838	27	5068
gl	Green Lantern	325582	69	4718

5.2. Box Office Data

The other piece of information needed for the experiments is the weekly movie sales. *The Numbers*[7] offers daily U.S. box office information of the top 50 movies being screened on that day. Among other information, this webpage offers the daily gross revenue, difference with respect to the previous day, number of theaters in which the film is screened, number of days since release and the total gross revenue. We scraped this data on a daily basis. As was the case for the stock market, instead of directly working with the gross revenue, we work with the logarithmic returns of the series.

5.3. Summary Tree

Similar to the stock market application, below is a representation of the summary tree for the Box office application.

The tree shows that adding sentiment information of the tweets does not help forecasting the box office revenue. In contrast, the use of the Twitter volume information does result in better predictions when using linear regression or SVMs with a lag greater than two days. It should be noted that the forecasting period for this domain is much shorter than for the stock market and thus these results may not be as robust as the ones in Section 4.

[6] `http://www.imdb.com/movies-in-theaters/`

[7] `http://www.the-numbers.com/charts/today.php`

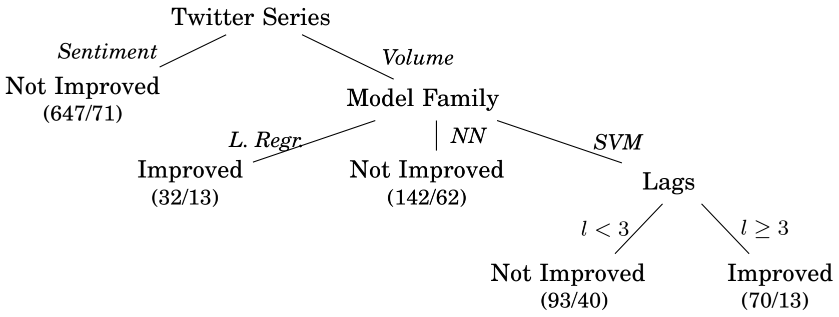

Fig. 5. Summary tree for the movie box office application.

5.4. Case Study: Green Lantern

Among the eight films for which we performed the experiments we have selected *Green Lantern* to be discussed in-depth. The results in this section correspond to the experiments using the non-LDA-filtered version of the tweets dataset and the *C-Flm* sentiment classifier.

Figure 6 shows the graphical representation of the gross revenue and tweet volume time series for the period commencing from 17 June to 23 August 2011. *Green Lantern* was released on 16 June in the U.S. As can be seen in the figure, the volume and the gross are seemingly closely related.

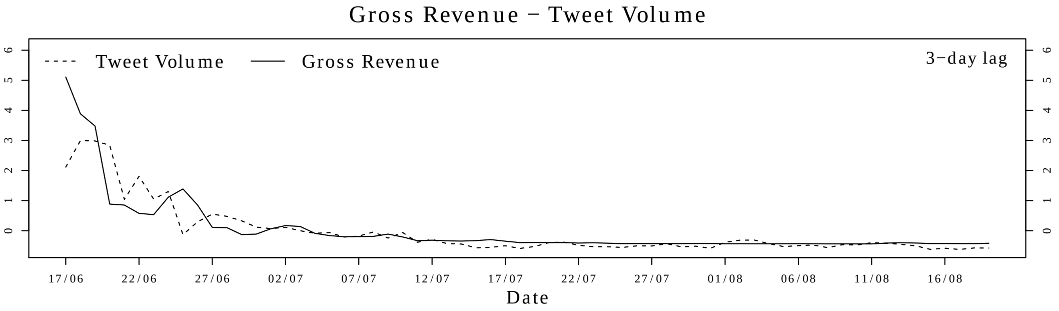

Fig. 6. Gross revenue versus daily tweet volume for Green Lantern

As in the previous application, we perform the statistical tests to determine the relation between the time series.

Nonlinearity. Teräsvirta's neural network test for neglected nonlinearity yielded the p-values shown Table XIII. The results clearly point to a nonlinear relationship between volume and gross revenue, indicated by a strong reject of the null hypothesis of linearity.

Table XIII.

Gross → Volume	Volume → Gross	Gross → Sentiment	Sentiment → Gross
2.94e-07	**0.0205**	0.6783	0.4865

Causality. The causality tests results suggest that there is causality between the daily volume of messages and the gross revenue. As for the causality between the sentiment index and the gross revenue the tests are not conclusive.

Table XIV.

Lag	Gross → Volume	Volume → Gross	Gross → Sentiment	Sentiment → Gross
1	0.0420	0.1124	0.6791	0.7799
2	1.30e-06	4.08e-05	0.8827	0.4080
3	1.75e-05	1.46e-15	0.4062	0.5050
4	6.73e-07	1.96e-15	0.6346	0.0185

Model fitting and evaluation. Table XV shows the difference in accuracies when the Tweet Volume is added to the prediction using linear regression of *Green Lantern*'s box office. Accuracies increase for lags between 2 and 4 although the Kappa statistic is still very low after adding the index. This is due to the short lifespan of movies which results in a small amount of data.

Table XV. Performance of linear regression models for the prediction of daily gross adding daily tweet volume.

Lag	Acc w/o	Acc w/	Kappa w/o	Kappa w/	Lag	Acc w/o	Acc w/	Kappa w/o	Kappa w/
1	0.3593	0.3281	-0.2964	-0.3596	3	0.4354	0.5322	-0.0753	0.0760
2	0.4603	0.5079	-0.0200	0.0542	4	0.4098	0.5081	-0.1273	0.0358

In the same way, the difference between the evaluation metrics displayed in Table XVI are for the case where we use the Twitter sentiment index for predicting with an SVM with a sigmoid kernel. The accuracy improves for the same lags as the previous experiment.

Table XVI. Performance of SVMs with a sigmoid kernel for the prediction of daily gross with Twitter sentiment.

Lag	Acc w/o	Acc w/	Kappa w/o	Kappa w/	Lag	Acc w/o	Acc w/	Kappa w/o	Kappa w/
1	0.6875	0.6093	0.3694	0.1951	3	0.4193	0.6129	-0.1797	0.2085
2	0.5238	0.5238	0.0606	0.0415	4	0.4666	0.5333	-0.0561	0.0572

6. CONCLUSIONS AND FUTURE WORK

In this work we set out to assess whether the addition of public information extracted from Twitter is of any use for obtaining better time series predictions. A large number of studies have been published on the usage of data derived from social networks for improving predictions, mostly on the subject of stock market and sales. Therefore we selected these two areas as case studies to be tested under a unified framework. This required the integration of several techniques from many applied disciplines: natural language processing, statistics, machine learning, among others. In particular, the implementation of a decision tree allowed us to orderly summarize the observations and group the experiments into different sets of parameters that lead to similar results. Using the summary tree as a guide we extracted those instances of model and target that make best use of Twitter data, and we arrived to the following general conclusions. For the stock market application we found that non linear models (SVMs and neural networks) with the addition of Twitter information, either in the form of number of tweets (volume) or a public sentiment index, do improve as predictors of the trend of volatility indices (e.g. VXO, VIX) and historic volatilities of stocks. The case of predicting the trend of benefits of particular stocks or indices (e.g. AAPL or OEX) is more dependent on the parameters, the input data and the Twitter classifier; in fact, the most relevant successful case is the SVM with poly-2 kernel for predicting OEX with the sentiment index obtained with the stock-specific classifier (*C-Stk*). In more detail, SVMs with sigmoid kernel far succeed other variants of this family of machine models in predicting volatility indices with either Twitter index (sentiment or volume), whereas Neural Networks outperformed in predicting volatility when using Twitter volume and lag ≥ 3 of past information of the target series. On the other hand, linear regression models are not able to exploit any of the Twitter derived indices. In the movie revenue application using the Sentiment index does not improve the predictions at all. However, using Tweet Volume results in an increase of forecasting performance in the cases of SVMs and linear models.

Future work. In this work we have proposed a robust methodology for the study of whether the addition of Twitter data helps in forecasting in two scenarios. Naturally, our framework could be improved in many ways, we proceed by explaining a few of the ideas we have not implemented yet. Clearly, the Twitter data collection and preprocessing is one of the crucial points of our framework. Therefore, many of the improvements go along the line of improving the sentiment classifier and index. For example, we are currently training a single sentiment classifier independently of the application domain. Tuning the sentiment classifier to the particularities of each of the application domains could probably result in better sentiment indices. Additionally, one could use the geographic location of the tweets to include those that are relevant to the task at hand (e.g. if we predict the sales in the US of a certain movie, then we should use tweets from the US only). The problem, however, of the approaches that filter out hurtful tweets, is the lack of enough labelled data to train the sentiment classifiers.

We could also improve the datasets used for training the sentiment classifier. For example, we noticed that happy smileys are much more common than sad ones. Thus, in order to train a sentiment classifier with a balanced dataset, the total number of instances is limited by the negative smileys. Co-training [Blum and Mitchell 1998] might alleviate this problem by labelling additional tweets that do not contain smileys, thus being able to train sentiment classifiers with much more data. Finally, the sentiment index is computed as a simple percentage of daily messages written on a given topic, a smarter way of building the index is for instance to weigh each tweet by its potential audience, e.g. the number of followers of the author. The idea is that users with many followers are bound to be more influential than others with fewer followers.

A restriction of our current methodology is to limit the predictions to the direction of the time series, we could include regression models in order to predict the actual values of the time series. Another improvement would be to consider ensembles of forecasting models as opposed to single models, given the recent popularity and success of this approach in many practical tasks [Cesa-Bianchi and Lugosi 2006; Rokach 2010].

In summary, we have presented a framework that allows us to study in a rigorous manner the question of whether Twitter helps in forecasting.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the fruitful comments of the anonymous referees that lead to an improvement of the original manuscript.

REFERENCES

AHKTER, J. AND SORIA, S. 2010. Sentiment analysis: Facebook status messages. *nlp.stanford.edu*, 1–19.

ASUR, S. AND HUBERMAN, B. A. 2010. Predicting the future with social media. *CoRR abs/1003.5699*.

BIFET, A. AND FRANK, E. 2010. Sentiment knowledge discovery in Twitter streaming data. In *Discovery Science*. Springer, 1–15.

BIFET, A., HOLMES, G., KIRKBY, R., AND PFAHRINGER, B. 2010. Moa: Massive online analysis. *The Journal of Machine Learning Research 11*, 1601–1604.

BLEI, D. M., NG, A. Y., AND JORDAN, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research 3,* 4-5, 993–1022.

BLUM, A. AND MITCHELL, T. 1998. Combining labeled and unlabeled data with co-training. Morgan Kaufmann Publishers, 92–100.

BOLLEN, J., MAO, H., AND ZENG, X. 2011. Twitter mood predicts the stock market. *J. of Computational Science 2,* 1, 1–8.

CESA-BIANCHI, N. AND LUGOSI, G. 2006. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA.

COHEN, J. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement 20,* 1, 37–46.

CULOTTA, A. 2010. Detecting influenza outbreaks by analyzing Twitter messages. *CoRR abs/1007.4748*.

DAS, S. AND CHEN, M. 2001. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *In Asia Pacific Finance Association Annual Conf. (APFA)*.

DIKS, C. AND PANCHENKO, V. 2006. A new statistic and practical guidelines for nonparametric Granger causality testing. *J. of Economic Dynamics & Control 30*, 1647–1669.

GAMA, J. A., SEBASTIÃO, R., AND RODRIGUES, P. P. 2009. Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*. ACM Press, New York, USA, 329.

GO, A., BHAYANI, R., AND HUANG, L. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*.

GRANGER, C. W. J. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica 37,* 3, 424–38.

GRUHL, D., GUHA, R., KUMAR, R., NOVAK, J., AND TOMKINS, A. 2005. The predictive power of online chatter. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. KDD '05. ACM, New York, NY, USA, 78–87.

HASTIE, T., TIBSHIRANI, R., AND FRIEDMAN, J. H. 2003. *The Elements of Statistical Learning* Corrected Ed. Springer.

HOFFMAN, M. D., BLEI, D. M., AND BACH, F. 2010. Online learning for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems 23*, 856–864.

LAMPOS, V., BIE, T. D., AND CRISTIANINI, N. 2010. Flu detector - Tracking epidemics on Twitter. In *ECML/PKDD (3)*. 599–602.

LEE, T.-H., WHITE, H., AND GRANGER, C. W. 1993. Testing for neglected nonlinearity in time series models: A comparison of neural network methods and alternative tests. *J. of Econometrics 56,* 3, 269–290.

MISHNE, G. AND GLANCE, N. 2005. Predicting movie sales from blogger sentiment.

MITCHELL, T. M. 1997. *Machine learning*. McGraw Hill series in computer science. McGraw-Hill.

O'CONNOR, B., BALASUBRAMANYAN, R., ROUTLEDGE, B. R., AND SMITH, N. A. 2010. From tweets to polls: linking text sentiment to public opinion time series. In *Proceeding of the International AAAI Conference on Weblogs and Social Media*. Washington, DC, 122–129.

PANG, B. AND LEE, L. 2008. Opinion mining and sentiment analysis. Ph.D. thesis.

PETROVIĆ, S., OSBORNE, M., AND LAVRENKO, V. 2010. The Edinburgh Twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*. Association for Computational Linguistics, 25–26.

POLANYI, L. AND ZAENEN, A. 2006. Contextual valence shifters. *Computing attitude and affect in text: Theory and applications 20*, 1–10.

POTTS, C. 2010. On the negativity of negation. In *Proceedings of Semantics and Linguistic Theory*. Vol. 20.

RITTERMAN, J., OSBORNE, M., AND KLEIN, E. 2009. Using prediction markets and Twitter to predict a swine flu pandemic. In *Proceedings of the 1st International Workshop on Mining Social Media*.

ROKACH, L. 2010. Ensemble-based classifiers. *Artif. Intell. Rev. 33,* 1-2, 1–39.

TERÄSVIRTA, T., LIN, C.-F., AND GRANGER, C. W. J. 1993. Power of the neural network linearity test. *J. of Time Series Analysis 14*, 209–220.

TODA AND YAMAMOTO. 1995. Statistical inferences in vector autoregressions with possibly integrated processes. *J. Econometrics 66*, 225–250.

TSAY, R. S. 2010. *Analysis of Financial Time Series* 3rd Ed. Wiley.

TUMASJAN, A., SPRENGER, T., SANDNER, P., AND WELPE, I. 2010. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*. 178–185.

TWITTER. 2010a. Developer rules of the road. http://webarchive.nationalarchives.gov.uk/20100520095247/dev.twitter.com/pages/api_terms. [Archived on 20-May-2010].

TWITTER. 2010b. Draft: Twitter rules for API rules. http://webarchive.nationalarchives.gov.uk/20100409154700/http://twitter.com/apirules. [Archived on 09-Apr-2010].

TWITTER. 2012. Twitter Translation Center adds right-to-left languages. http://blog.twitter.com/2012/01/twitter-translation-center-adds-right.html. [Posted on 25-Jan-2012].

WAKAMIYA, S., LEE, R., AND SUMIYA, K. 2011. Crowd-powered TV viewing rates: Measuring relevancy between tweets and TV programs. In *DASFAA Workshops*. 390–401.

WHITE, H. 1989. An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks. In *Neural Networks, 1989. IJCNN., International Joint Conference on*. IEEE, 451–455.

WOLFRAM, M. S. A. 2010. Modelling the stock market using Twitter. M.S. thesis, School of Informatics, University of Edinburgh.

ZHANG, X., FUEHRES, H., AND GLOOR, P. A. 2010. Predicting stock market indicators through Twitter "I hope it is not as bad as I fear". In *Collaborative Innovation Networks (COINs)*. Savannah, GA. (to appear in Procedia - Social and Behavioral Sciences. http: www.elsevier.com/locate/).

Received February 2012; revised March 2012; accepted June 2012

APPENDIX

A. STOCK MARKET SUMMARY TREE

5-level decision tree showcasing the parameter settings that allow for an accuracy improvement of at least 5% when adding Twitter information to the model. Accuracy must also surpass 50%. Paths that lead to a TRUE-tagged leaf describe models that improve with Twitter data.

```
Model = glm : FALSE (2418/64) [1220/39]
Model = nnet−2 : FALSE (2375/220) [1221/131]
Model = nnet−3
|   Twitter Series = Both : FALSE (761/60) [395/38]
|   Twitter Series = Sentiment : FALSE (1059/87) [486/37]
|   Twitter Series = Volume
|   |   Lags < 4.5 : FALSE (492/78) [258/39]
|   |   Lags ≥ 4.5
|   |   |   Target Series = Close : FALSE (62/9) [23/3]
|   |   |   Target Series = Volatility
|   |   |   |   Symbol = goog : TRUE (4/0) [2/0]
|   |   |   |   Symbol = gspc : FALSE (13/3) [11/5]
|   |   |   |   Symbol = msft : TRUE (2/1) [4/3]
|   |   |   |   Symbol = oex : FALSE (18/3) [5/0]
|   |   |   |   Symbol = yhoo : TRUE (4/1) [2/2]
|   |   |   |   Symbol = aapl : FALSE (4/0) [2/0]
Model = nnet−4
|   Twitter Series = Both : FALSE (755/68) [390/37]
|   Twitter Series = Sentiment : FALSE (1024/90) [509/44]
|   Twitter Series = Volume
|   |   Lags < 2.5 : FALSE (239/23) [124/14]
|   |   Lags ≥ 2.5
|   |   |   Symbol = goog : FALSE (23/2) [6/0]
|   |   |   Symbol = gspc : FALSE (76/27) [40/10]
|   |   |   Symbol = msft : FALSE (20/7) [7/3]
|   |   |   Symbol = oex : FALSE (72/17) [44/14]
|   |   |   Symbol = vix
|   |   |   |   Lags < 3.5 : TRUE (22/4) [12/5]
|   |   |   |   Lags ≥ 3.5 : FALSE (51/17) [20/1]
|   |   |   Symbol = vxo : FALSE (83/24) [44/22]
|   |   |   Symbol = yhoo : FALSE (17/2) [9/0]
|   |   |   Symbol = aapl : FALSE (21/1) [10/1]
Model = nnet−5 : FALSE (2442/316) [1188/144]
Model = svm−poly−2
|   Twitter Dataset = goog : FALSE (294/1) [146/2]
|   Twitter Dataset = googlda : FALSE (306/1) [168/0]
|   Twitter Dataset = msft : FALSE (331/0) [146/0]
|   Twitter Dataset = msftlda : FALSE (316/1) [157/0]
|   Twitter Dataset = yhoo : FALSE (310/26) [156/13]
|   Twitter Dataset = yhoolda
|   |   Sent. Classifier  = Cl−En : FALSE (98/2) [48/0]
|   |   Sent. Classifier  = Cl−Ml : FALSE (97/1) [49/0]
|   |   Sent. Classifier  = Cl−Stk
|   |   |   Symbol = gspc : FALSE (10/3) [5/1]
|   |   |   Symbol = oex : TRUE (8/4) [2/1]
|   |   |   Symbol = vix : FALSE (16/6) [7/1]
|   |   |   Symbol = vxo : TRUE (15/7) [12/4]
|   |   |   Symbol = yhoo : FALSE (18/5) [7/2]
|   |   Sent. Classifier = OpinionFinder : FALSE (33/0) [17/0]
|   Twitter Dataset = aapl : FALSE (288/12) [148/6]
|   Twitter Dataset = aapllda : FALSE (313/2) [159/0]
Model = svm−poly−3 : FALSE (2481/79) [1199/27]
Model = svm−poly−4
|   Twitter Dataset = goog : FALSE (283/3) [157/0]
|   Twitter Dataset = googlda : FALSE (316/1) [158/0]
|   Twitter Dataset = msft : FALSE (318/0) [159/0]
|   Twitter Dataset = msftlda : FALSE (327/0) [146/1]
|   Twitter Dataset = yhoo
|   |   Sent. Classifier = Cl−En : FALSE (92/0) [55/0]
|   |   Sent. Classifier = Cl−Ml : FALSE (101/4) [39/1]
|   |   Sent. Classifier = Cl−Stk
|   |   |   Target Series = Close
|   |   |   |   Twitter Series = Both : FALSE (19/6) [6/1]
|   |   |   |   Twitter Series = Sentiment : FALSE (18/5) [7/2]
|   |   |   |   Twitter Series = Volume : TRUE (8/1) [3/1]
|   |   |   Target Series = Volatility : FALSE (50/4) [19/2]
|   |   Sent. Classifier = OpinionFinder : FALSE (34/3) [15/2]
|   Twitter Dataset = yhoolda
|   |   Sent. Classifier = Cl−En : FALSE (94/2) [52/0]
|   |   Sent. Classifier = Cl−Ml : FALSE (88/0) [58/1]
|   |   Sent. Classifier = Cl−Stk
|   |   |   Symbol = gspc : FALSE (10/2) [5/2]
|   |   |   Symbol = oex : TRUE (7/3) [3/2]
|   |   |   Symbol = vix : FALSE (18/4) [5/3]
|   |   |   Symbol = vxo : TRUE (18/8) [9/3]
|   |   |   Symbol = yhoo : FALSE (21/7) [4/0]
|   |   Sent. Classifier = OpinionFinder : FALSE (41/0) [9/0]
|   Twitter Dataset = aapl : FALSE (287/12) [149/6]
Model = svm−poly−5 : FALSE (2431/72) [1249/34]
Model = svmradial : FALSE (2082/59) [1081/35]
Model = svmsigmoid
|   Target Series = Close
|   |   Symbol = goog
|   |   |   Twitter Series = Both : FALSE (21/0) [9/0]
|   |   |   Twitter Series = Sentiment : FALSE (25/0) [15/1]
|   |   |   Twitter Series = Volume : TRUE (10/1) [5/1]
|   |   Symbol = gspc : FALSE (230/43) [109/17]
|   |   Symbol = msft : FALSE (57/5) [30/1]
|   |   Symbol = oex : FALSE (206/50) [109/35]
|   |   Symbol = vix : FALSE (210/36) [101/20]
|   |   Symbol = vxo
|   |   |   Lags < 2.5
|   |   |   |   Lags < 1.5 : FALSE (51/7) [24/2]
|   |   |   |   Lags ≥ 1.5 : TRUE (49/21) [26/8]
|   |   |   Lags ≥ 2.5 : FALSE (154/14) [66/11]
|     |     Symbol = yhoo : FALSE (60/15) [27/3]
|     |     Symbol = aapl : FALSE (51/19) [33/9]
|     Target Series = Volatility  :  FALSE (1227/90) [615/39]
```

B. BOX OFFICE SUMMARY TREE

As with the stock market case, below is the decision tree showcasing the parameter settings that allow for an accuracy improvement of at least 5% when adding Twitter information to the model. Accuracy must also surpass 50%. Variations of the same model families have been aggregated.

```
Twitter Series = Sentiment : FALSE (647/71) [299/39]
Twitter Series = Volume
|     Model = glm : TRUE (32/13) [21/7]
|     Model = nnet : FALSE (142/62) [71/25]
|     Model = svm
|     |     Lags < 2.5 : FALSE (93/40) [65/26]
|     |     Lags ≥ 2.5 : TRUE (70/13) [36/16]
```