Estimating the Time between Twitter Messages and Future Events

Ali Hürriyetoğlu, Florian Kunneman, and Antal van den Bosch
Centre for Language Studies, Radboud University Nijmegen
P.O. Box 9103
NL-6500 HD Nijmegen
ali.hurriyetoglu@gmail.com, {f.kunneman,a.vandenbosch}@let.ru.nl

ABSTRACT

We describe and test three methods to estimate the remaining time between a series of microtexts (tweets) and the future event they refer to via a hashtag. Our system generates hourly forecasts. A linear and a local regression-based approach are applied to map hourly clusters of tweets directly onto time-to-event. To take changes over time into account, we develop a novel time series analysis approach that first derives word frequency time series from sets of tweets and then performs local regression to predict time-to-event from nearest-neighbor time series. We train and test on a single type of event, Dutch premier league football matches. Our results indicate that in an 'early' stage, four days or more before the event, the time series analysis produces time-to-event predictions that are about one day off; closer to the event, local regression attains a similar accuracy. Local regression also outperforms both mean and median-based baselines, but on average none of the tested system has a consistently strong performance through time.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Spatial-Temporal Systems

General Terms

Algorithms, Performance

Keywords

Time series analysis, Event prediction, Twitter

1. INTRODUCTION

With the advent of social media, data streams of unprecedented volume have become available. These streams do not only contain text, but also identity markers of the persons who generated the text, and the time at which the messages were published. The availability of massive amounts of time-stamped texts is an invitation to incorporate time series analysis methods into the natural language processing toolbox. For instance, predictive models can be built through time series analysis that can estimate the likelihood and time of future events.

Our study focuses on textual data published by humans via social media about particular events. If the starting point of the event in time is taken as the anchor $t = 0$ point in time, texts can be viewed in relation to this point, and generalizations can be made over texts at different distances in time to $t = 0$. The goal of this paper is to present new methods that are able to automatically estimate the time-to-event from a stream of microtext messages. These methods could serve as modules in news media mining systems[1] to fill upcoming event calendars. The methods should be able to work robustly in a stream of messages, and the dual goal would be to make (i) reliable predictions of times-to-event (ii) as early as possible. Predicting that an event is starting imminently is arguably less useful than being able to predict its start in a number of days. This implies that if a method requires a sample of tweets (e.g. with the same hashtag) to be gathered during some time frame, the frame should not be too long, otherwise predictions could come in too late to be relevant.

In this paper we test the predictive capabilities of three different approaches. The first system is based on linear regression and maps sets of tweets with the same hashtag during an hour to a time-to-event estimate. The second system attempts to do the same based on local regression. The third system uses time series analysis. It takes into account more than a single set of tweets: during a certain time period it samples several sets of tweets in fixed time frames, and derives time series information from individual word frequencies in these samples. It compares these word frequency time series profiles against a labeled training set of profiles in order to find similar patterns of change in word frequencies. The method then adopts local regression: finding a nearest-neighbor word frequency time series, the time-to-event stored with that neighbor is copied to the tested time series. With this third system, and with the comparison against the second system, we can test the hypothesis that it is useful to gather time series information (more specifically, patterns in word frequency changes) over an amount of time.

This paper is structured as follows. We describe the relation of our work to earlier research in Section 2. The three systems are described in Section 3. Section 4 describes the overall experimental setup, including a description of the data, the baseline, and the evaluation method used. The results are presented and analyzed in Section 5. We conclude with a discussion of the results and future studies in Section 6.

[1]For instance, `http://www.zapaday.com/`

2. RELATED RESEARCH

The growing availability of digital texts with time stamps, such as e-mails, weblogs, and online news, has spawned various types of studies on the analysis of patterns in texts over time. An early publication on the general applicability of time series analysis on time-stamped text is [2]. A more recent overview of future predictions using social media is [5]. A popular goal of time series analysis of texts is *event prediction*, where a correlation is sought between a point in the future and preliminary texts.

Ritter *et al.* train on annotated open-domain event mentions in tweets in order to create a calendar of events based on explicit date mentions and words typical of the event [3]. While we also aim to estimate the point in time at which an event will take place, our focus lies on the pattern of anticipation seen in tweets linked to the time until the event occurs rather than specific time references to a future event. [4] do look at anticipation seen in tweets, but focus on personal activities in the very near future, while we aim to predict the time-to-event of potentially large-scale news events as early as possible.

3. METHODS

In this section we introduce the methods adopted in our study. They operate on streams of tweets, and generate hourly forecasts for the events that tweets with the same hashtag refer to. The single tweet is the smallest unit available for this task; we may also consider more than one tweet and aggregate tweets over a certain time frame. If these single tweets or sets of tweets are represented as bag-of-words vectors, the task can be cast as a regression problem: mapping a feature vector onto a continuous numeric output representing the time-to-event. In this study the smallest time unit is one hour, and all three methods work with this time frame.

3.1 Linear and local regression

In linear regression, each feature in the bag-of-words feature vector (representing the presence or frequency of occurrence of a specific word) can be regarded as a predictive variable to which a weight can be assigned that, in a simple linear function, multiplies the value of the predictive variable to generate a value for the response variable, the time-to-event. A multiple linear regression function can be approximated by finding the weights for a set of features that generates the response variable with the smallest error.

Local regression, or local learning [1], is the numeric variant of the k-nearest neighbor classifier. Given a test instance, it finds the closest k training instances based on a similarity metric, and bases a local estimation of the numeric output by taking some average of the outcomes of the closest k training instances.

Linear regression and local regression can be considered baseline approaches, but are complementary. While in linear regression an overall pattern is generated to fit the whole training set, local regression only looks at local information for classification (the characteristics of single instances). Linear regression is unfit for approximating gaussian or other non-linear distributions; as we will see, there are reasons to believe that there are substantial differences in tweets posted in different periods of time before an event. In contrast, local regression is unbiased and will adapt to any local distribution.

3.2 Time series analysis

Time series are data structures that contain multiple measurements of data features over time. If values of a feature change meaningfully over time, then time series analysis can be used to capture this pattern of change. Comparing new time series with memorized time series can reveal similarities that may lead to a prediction of a subsequent value or, in our case, the time-to-event. Our time series approach extends the local regression approach by not only considering single sets of aggregrated tweets in a fixed time frame (e.g. one hour in our study), but creating sequences of these sets representing several consecutive hours of gathered tweets. Using the same bag-of-words representation as the local regression approach, we find nearest neighbors of sequences of bag-of-word vectors rather than single hour frames. The similarity between a test time series and a training time series of the same length is calculated by computing their Euclidean distance. In this study we did not further optimize any hyperparameters; we set $k = 1$.

The time series approach generates predictions by following the same strategy as the simple local regression approach: upon finding the nearest-neighbor training time series, the time-to-event of this training time series is taken as the time-to-event estimate of the test time series. In case of equidistant nearest neighbors, the average of their associated time-to-events is given as the prediction.

4. EXPERIMENTAL SET-UP

4.1 Data collection

For this study we chose football matches as a specific type of event. They occur frequently, have a distinctive hashtag by convention ('#ajafey' for a match between Ajax and Feyenoord) and often generate a useful amount of tweets: up to tens of thousands of tweets per match. For the collection of training and test data we focused on Dutch football matches played in the *Eredivisie*. We harvested tweets by means of `twiqs.nl`, a database of Dutch tweets from December 2010 onwards. We selected the (arbitrary) top 6 teams of the league[2], and queried all matches played between them in 2011 and 2012. For each query, the conventional hashtag for a match was used with a restricted search space of three weeks before the time of the match until the start time of the match (to ensure that the collected tweets were referring to that specific match, and not to an earlier match consisting of the same home and away team and therefore the same hashtag).

The queries resulted in tweets referring to 60 matches between the selected six teams in the period from January 2011 until December 2012. From these, we selected the matches with the most frequent similar starting time, Sundays at 2:30 PM, for our experiment. As we focused on the amount of hours before an event, the actual time when a tweet is posted (for example during the night or in the afternoon) can bias the type of tweet; with the fixed starting time this

[2]Ajax, Feyenoord, PSV, FC Twente, AZ Alkmaar and FC Utrecht

effect is neutralized. To generate training and test events that simulate a system trained on passed events and tested on upcoming events, we selected tweets referring to matches played in 2011 (a calendar year comprising two halfs of a football season) as training data and tweets referring to 2012 matches as test data. This resulted in 12 matches as training events (totaling 54,081 tweets) and 14 matches as test events (40,204 tweets).

The time-to-event in hours was calculated for every tweet, based on their time of posting and the known start time of the event they referred to. For this task we did not take tweets into account that were posted during and after matches. We also constrained the number of days before the event: for both training and test sets, tweets were kept within eight days before the event. Although this is an artificial constraint, the eight days window captures the vast majority, about 98%, of forward-looking tweets.

4.2 Generation of training and test data

The goal of the experiments was to compare systems that generate hourly forecasts of the event start time for each test event. This was done based on the information in aggregated sets of tweets within the time span of an hour. Aggregation is done by treating all training events as one collection during the extraction of features. The linear and local regression methods only operate on vectors representing hour blocks. The time series analysis approach makes use of longer sequences of six hour blocks - this number was empirically set in preliminary experiments.

The aggregated tweets were used as training instances for the linear and local regression methods. To maximize the number of training instances, we generated a sequence of overlapping instances using the minute as a finer-grained shift unit. At every minute, all tweets posted within the hour before the tweets in that minute were added to the instance.

In order to reduce the feature space for the linear and local regression instances, we pruned every bag-of-word feature that occured less than 500 times in the training set. Linear regression was applied by means of R[3]. Absolute occurrence counts of features were taken into account. For local regression we made use of the k-NN implementation as part of TiMBL[4], setting $k = 5$, using Information Gain feature weighting, and an overlap-based metric as similary metric that does not count matches on zero values (features marking words that are absent in both test and training vectors). For k-NN, the binary value of features were used.

The time series analysis vectors are not filled with absolute occurrence counts, but with relative and smoothed frequencies. After having counted all words in each time frame, two frequencies are computed for each word. The first, the overall frequency of a word, is calculated as the sum of its counts in all time frames, divided by the total number of tweets in all time frames in our 8-day window. This frequency ranges between 0 (the word does not occur) and 1 (the word occurs in every tweet). The second frequency is computed per time frame for each word, where the word count in that frame is divided by the number of tweets in the frame. The latter frequency is the basic element in our time series calculations.

As many time frames contain only a small number of tweets, especially the frames more than a few days before the event, word counts are sparse as well. Besides taking longer time frames of more than a single sample size, frequencies can also be smoothed through typical time series analysis smoothing techniques such as moving average smoothing. We apply a pseudo-exponential moving average filter by replacing each word count by a weighted average of the word count at time frames t, $t-1$, and $t-2$, where $w_t = 4$ (the weight at t is set to 4), $w_{t-1} = 2$, and $w_{t-2} = 1$.

4.3 Evaluation and baselines

A common metric for evaluating numeric predictions is the Root Mean Squared Error (RMSE), cf. Equation 1. For all hourly forecasts made in N hour frames, a sum is made of the squared differences between the actual value v_i and the estimated value e_i; the (square) root is then taken to produce the RMSE of the prediction series.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i - e_i)^2} \qquad (1)$$

We computed two straightforward baselines derived from the training set: the median and the mean of time-to-event over all training tweets. For the median baseline, all tweets in the training set were ordered in time and the median time was identified. As we use one-hour time frames throughout our study, we round the median by the one-hour time frame it is in, which turns out to be -3 hours. The mean is computed by averaging the time-to-event of all tweets, and again rounded at the hour. The mean is -26 hours.

5. RESULTS

Table 1 displays the averaged RMSE results on the 14 test events. On average the performance of the linear regression method is worse than both baselines, while the time series analysis outperforms the median baseline. Given that the best performing method is still an unsatisfactory 43 hours off, there is still a lot of improvement needed. The best method per event varies. Even linear regression, which has a below baseline performance on average, leads to the best RMSE for two events. It appears that some negative deviations (110 for 'twefey', 410 for 'tweaja') lead to the poor average RMSE.

The average performance of the different methods in terms of their RMSE according to hourly forecasts is plotted in Figure 1. In the left half of the graph the three systems outperform the baselines, except for an error peak of the linear regression method at around $t = -150$. Before $t = -100$ the time series prediction is performing rather well, with RMSE values averaging 23 hours. The linear regression and local regression methods produce larger errors at first, decreasing as time progresses. In the second half of the graph, however, only the local regression method retains fairly low RMSE

[3]http://www.r-project.org/
[4]`http://ilk.uvt.nl/timbl`

	Spring 2012								Fall 2012						
	azaja	feyaz	feyutr	psvfey	tweaja	twefey	tweutr	utraz	azfey	psvaz	twefey	utraz	utrpsv	utrtwe	Av (sd)
Baseline Median	63	49	54	62	38	64	96	71	62	67	62	66	61	62	63 (12)
Baseline Mean	51	**40**	44	51	**31**	52	77	58	**50**	55	51	53	49	**51**	51 (10)
Linear regression	52	42	59	54	410	**41**	41	33	111	**31**	110	54	37	68	82 (94)
Local regression	**48**	44	**35**	**41**	43	43	**31**	**20**	57	40	52	**48**	**34**	52	**43** (9)
Time Series	**48**	50	42	43	45	**41**	63	70	48	58	**46**	71	59	63	54 (10)

Table 1: Overall Root Mean Squared Error scores for each method: difference in hours between the estimated time-to-event and the actual time-to-event

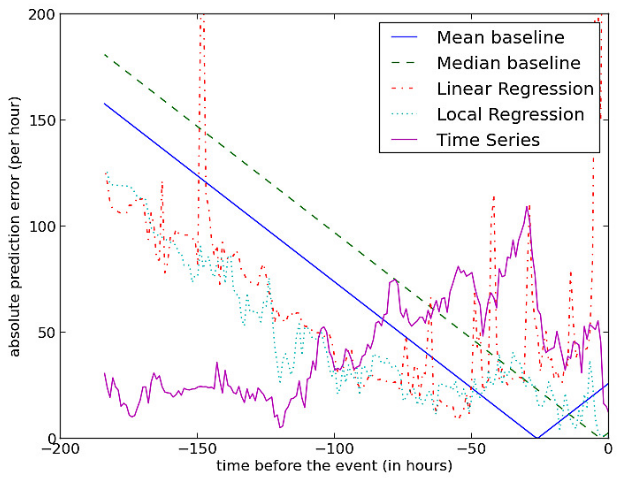

Figure 1: RMSE curves for the two baselines and the three methods for the last 192 hours before $t = 0$.

values at an average of 21 hours, while the linear regression method becomes increasingly erratic in its predictions. The time series analysis method also produces considerably higher RMSE values in the last days before the events.

6. CONCLUSION

In this study we explored and compared three approaches to time-to-event prediction on the basis of streams of tweets. We tested on the prediction of the time-to-event of football matches by generating hourly forecasts. When the three approaches are compared to two simplistic baselines based on the mean and median of the time-to-event of tweets sent before an event, only local regression displays better overall RMSE values on the tested prediction range of $192 \ldots 0$ hours before the event. Linear regression generates some highly erratic predictions and scores below both baselines. A novel time series approach that implements local regression based on sequences of samples of tweets performs better than the mean baseline, but under the median baseline.

Yet, the time series method generates fairly accurate forecasts during the first half of the test period. Before $t < -100$ hours, i.e. earlier than four days before the event, predictions by the time series method are only about a day off (23 hours on average in this time range). When $t \leq -100$, the local regression approach based on sets of tweets in hourly time frames is the better predictor, with RMSE values that are sometimes close to $t = 0$ (21 hours on average in this time range).

On the one hand, our results are not very strong: predictions that are more than two days off and that are at the same time only mildly better than simple baselines cannot be considered precise. However, the results indicate that if we divide the problem into an 'early' prediction system based on time series analysis and a 'late' prediction system based on local regression, we could limit the prediction error to within a day. If we can detect the point at which the time series analysis starts increasing its predicted time-to-event (which is the wrong trend as the event can only come closer in time), it is time to switch to the local regression system. In our data, this point is around $t = -100$.

In future work we plan to extend the current study in several directions. Most importantly, we plan to extend the study to other events, moving from football to other scheduled events, and from scheduled events to unscheduled events, the ultimate goal of a forecasting system like this. A second extension is to improve on the time series analysis method, particularly to investigate why it is performing well only up to several days before the future event (and what kind of patterns it matches successfully). We also plan to optimize the local regression approach, as we now utilize a fairly standard k-NN approach without optimized hyperparameters, and we have not optimized the selection of features either.

Acknowledgement

This study is financed by the COMMIT program as part of the Infiniti project.

7. REFERENCES

[1] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1–5):11–73, 1997.

[2] J. Kleinberg. Temporal dynamics of On-Line information streams. In *Data stream management: Processing high-speed data streams*. Springer, 2006.

[3] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 1104–1112. ACM, 2012.

[4] W. Weerkamp and M. De Rijke. Activity prediction: A Twitter-based exploration. In *Proceedings of TAIA'12*, Aug. 2012.

[5] S. Yu and S. Kak. A survey of prediction using social media. In *ArXiv e-prints*, Mar. 2012.