1 Method

Extract data collected from forums Timestamp, Author, Text Content. Using sliding window training method, group consecutive w posts together and perform regression on Δ_t. More formally, we are trying to learn a function f such that $f(\mathbf{x}_{t-w}, \ldots, \mathbf{x}_{t-1}) \approx \Delta_t$, where $\mathbf{x}_t$ is the feature vector of a post made at time t, and Δ_t is the time between the t-th post and the $(t-1)$-th post. The following are the features used:

Previous time differences All the time differences between posts made in the window. ($\mathbf{t}_\Delta$)

Time-based features Day of week, Hour of day. Provides contextual information about when the post was made. ($\mathbf{t}_{\text{ctx}}$)

Content features (text) Word frequency counts. Used regression to test effect of single regressor. Top F features are selected for extraction. ($\mathbf{v}$)

In the following experiments, the threads chosen from our extracted dataset are those with a 100 to 1000 posts. This amounted to 97 threads. The first 75% of the thread was used as training data, while the remaining 75% was used as test data. We used Support Vector machines for this regression task, employing a Radial Basis Function kernel as our learning algorithm.

The SVR module from the Python library scikit-learn was used in the implementation of this experiment.

1.1 Evaluation metrics

We use *Mean Absolute Percentage Error* (MAPE), to measure the performance of the learnt model. This value is given by

$$\frac{1}{N}\sum_{i=1}^{N}\left|\frac{A_i - F_i}{A_i}\right|$$

where A_i is the actual value, and F_i is the forecasted value for the instance i. Realistically, the model would not be able to come into contact with every possible window, since chances are it will make an error that causes it to visit a thread late, causing it to miss two posts or more. This value does not reflect how well the model will do in a real-time setting, but gives an idea of how far off the model is given a window.

We also want to know the *timeliness* of the model's visits. Yang et. al. [**?**] has a metric for measuring this. Taking Δt_i as the time difference between a post i and it's download time, the timeliness of the algorithm is given by

$$T = \frac{1}{N}\sum_{i=1}^{N}\Delta t_i$$

A good algorithm would give a low T-score. However, a crawler that hits the site repeatedly performs well according to this metric. The authors account for this by setting a bandwidth (fixed number of pages per day) for each iteration of their testing.

	MAPE	Pr_{miss}	Pr_{fa}	Pr_{error}	T-score	Posts	Visits
Average $w = 5$	330.285	0.951	0.054	0.502	6418.208	33.000	498.742
Average $w = 10$	305.557	0.955	0.053	0.504	4598.955	31.680	497.351
Average Δ_t	174.004	0.938	0.065	0.501	1764.474	34.000	574.031
$w = 5, \mathbf{t}_\Delta$	18.884	0.931	0.064	0.498	1541.595	33.000	547.062
$w = 5, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}$	18.885	0.931	0.064	0.498	1541.592	33.000	547.062
$w = 5, \mathbf{v}$	9.382	0.923	0.063	0.493	1597.533	33.000	545.495
$w = 5, \mathbf{v}, \mathbf{t}_\Delta$	18.877	0.931	0.064	0.498	1541.588	33.000	547.062

Table 1: Experiment results

In our experimental results, we also take into account the number of page requests made in comparison to the number of posts.

Viewing the posts made during the thread's lifetime as segmentations of the thread, and the visits made as hypotheses of where the segmentations are, we use the Pr_{error} metric from Georgescul et. al., 2006 as a measure of how close the predictions are to the actual posts.

2 Results

The results for experiments done with different combinations of the above specified features are shown in Table 1.

Overall average and window average perform significantly worst than the learnt models, as reflected in both the MAPE and the T-score. There is also a slight improvement in the Pr_{error} in the learnt models.

Taking into account the T-score and the number of visits together, would seem that $\mathbf{t}_\Delta$, features representing the previous time intervals, are important features when determining the next time interval. In the absence of these features, we observe that the T-score increases by about 1%. In this experiment, we use purely word frequency features. This gives only a slight improvent over not using them.

High values for Pr_{miss} and low for Pr_{fa}, are due to Pr_{miss} being conditioned on there being a post within the window. Since the posts come in bursts, visits are fairly periodic, and intervals between visits are larger than post bursts. When there are more posts than visits in windows with posts, we have higher Pr_{miss}

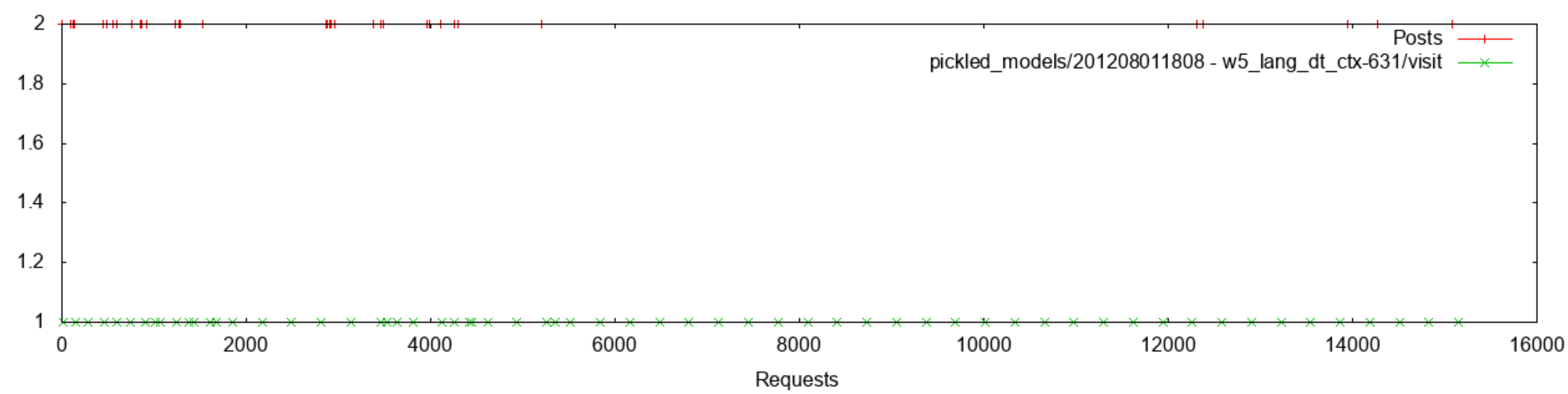

Figure 1: Visitation chart for a model using the $w = 5, \mathbf{t}_{\Delta}, \mathbf{t}_{\text{ctx}}, \mathbf{w}$ feature set. Invalid Predictions $= 0.758$, $Pr_{error} = 0.485$, T-score $= 119.612$, Posts $= 41$, Visits $= 62$

	MAPE	Pr_{miss}	Pr_{fa}	Pr_{error}	T-score	Posts	Visits
Average $w = 5$	330.285	0.951	0.054	0.502	6418.208	33.000	498.742
Average $w = 10$	305.557	0.955	0.053	0.504	4598.955	31.680	497.351
Average $w = 15$	308.547	0.954	0.054	0.504	3833.605	30.402	502.649
Average $w = 20$	265.124	0.953	0.054	0.504	3340.929	29.216	477.536
Average $w = 25$	257.844	0.955	0.052	0.503	3186.309	28.000	453.722
Average $w = 30$	244.988	0.957	0.050	0.504	2859.380	26.680	436.918
$w = 5, \mathbf{t}_\Delta$	18.884	0.931	0.064	0.498	1541.595	33.000	547.062
$w = 10, \mathbf{t}_\Delta$	19.647	0.937	0.061	0.499	1488.688	31.680	531.763
$w = 15, \mathbf{t}_\Delta$	20.195	0.939	0.061	0.500	1443.138	30.402	529.979
$w = 20, \mathbf{t}_\Delta$	20.220	0.938	0.059	0.499	1584.171	29.216	500.330
$w = 25, \mathbf{t}_\Delta$	20.953	0.937	0.056	0.496	1649.098	28.000	473.062
$w = 30, \mathbf{t}_\Delta$	21.242	0.941	0.054	0.498	1626.782	26.680	453.763
$w = 5, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}$	18.885	0.931	0.064	0.498	1541.592	33.000	547.062
$w = 10, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}$	19.647	0.937	0.061	0.499	1488.688	31.680	531.763
$w = 15, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}$	20.195	0.939	0.061	0.500	1443.138	30.402	529.979
$w = 20, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}$	20.220	0.938	0.059	0.499	1584.171	29.216	500.330
$w = 25, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}$	20.953	0.937	0.056	0.496	1649.098	28.000	473.062
$w = 30, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}$	21.242	0.941	0.054	0.498	1626.782	26.680	453.763
$w = 5, \mathbf{v}$	9.382	0.923	0.063	0.493	1597.533	33.000	545.495
$w = 10, \mathbf{v}$	13.863	0.934	0.061	0.498	1551.375	31.680	530.619
$w = 15, \mathbf{v}$	13.217	0.934	0.060	0.497	1507.589	30.402	528.247
$w = 20, \mathbf{v}$	14.849	0.930	0.059	0.494	1630.643	29.216	499.351
$w = 25, \mathbf{v}$	17.542	0.930	0.055	0.493	1700.990	28.000	472.031
$w = 30, \mathbf{v}$	18.627	0.937	0.054	0.496	1653.156	26.680	452.979
$w = 5, \mathbf{v}, \mathbf{t}_\Delta$	18.877	0.931	0.064	0.498	1541.588	33.000	547.062
$w = 10, \mathbf{v}, \mathbf{t}_\Delta$	19.645	0.937	0.061	0.499	1488.680	31.680	531.763
$w = 15, \mathbf{v}, \mathbf{t}_\Delta$	20.193	0.939	0.061	0.500	1443.130	30.402	529.979
$w = 20, \mathbf{v}, \mathbf{t}_\Delta$	20.220	0.938	0.059	0.499	1584.171	29.216	500.330
$w = 25, \mathbf{v}, \mathbf{t}_\Delta$	20.953	0.937	0.056	0.496	1649.098	28.000	473.062
$w = 30, \mathbf{v}, \mathbf{t}_\Delta$	21.242	0.941	0.054	0.498	1626.782	26.680	453.763

Table 2: Experiment results: Varying feature sizes

	MAPE	Pr_{miss}	Pr_{fa}	Pr_{error}	T-score	Posts	Visits
$w = 5, \mathbf{v}, \lvert\mathbf{v}\rvert = 5$	8.441	0.922	0.064	0.493	1601.321	33.000	545.371
$w = 5, \mathbf{v}, \lvert\mathbf{v}\rvert = 10$	8.632	0.924	0.064	0.494	1593.763	33.000	545.206
$w = 5, \mathbf{v}, \lvert\mathbf{v}\rvert = 15$	8.913	0.924	0.064	0.494	1594.276	33.000	545.381
$w = 5, \mathbf{v}, \lvert\mathbf{v}\rvert = 20$	9.382	0.923	0.063	0.493	1597.533	33.000	545.495
$w = 5, \mathbf{v}, \lvert\mathbf{v}\rvert = 25$	9.905	0.927	0.063	0.495	1597.295	33.000	545.619
$w = 5, \mathbf{v}, \lvert\mathbf{v}\rvert = 30$	10.836	0.925	0.063	0.494	1587.734	33.000	545.619

Table 3: Experiment results: Varying vocabulary size