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 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. ($\mathbf{w}$)

1.1 Evaluation metrics

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

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

The T*-score* metric measures the performance on a thread. This is the average time taken between a post being made and a visit made to retrieve that post. Limitations are that the value T-score does not factor in the number of times the page is hit. Keep track of the number of visits made as well. (Include explanation of T-score)

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 very poorly, on MAPE score, reflected in T-score.

Looking at T-score and no. of visits together, would seem that $\mathbf{t}_\Delta$ is important feature (Including reduces T-score).

Using content (word frequency) features for prediction, gives only slight improvement.

High values for Pr_{miss} and low for Pr_{fa}, are due to

Model	MAPE	Pr_{miss}	Pr_{fa}	Pr_{error}	T-score	Inv. pred	Posts	Visits
$w = 5, \mathbf{t}_\Delta$	19.404	0.932	0.063	0.498	1582.690	0.888	33.500	555.340
$w = 5, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}$	18.984	0.932	0.064	0.498	1596.708	0.889	33.419	561.839
$w = 5, \mathbf{w}$	9.786	0.926	0.062	0.494	1636.843	0.919	33.305	549.379
$w = 5, \mathbf{t}_\Delta, \mathbf{t}_{\text{ctx}}, \mathbf{w}$	19.225	0.933	0.062	0.498	1561.098	0.889	33.402	541.464
Average $w = 5$	332.502	0.954	0.052	0.503	6521.876	0.867	33.427	498.073
Average $w = 10$	186.303	0.941	0.060	0.500	1677.474	0.798	32.042	545.632
Average Δt	179.227	0.937	0.061	0.499	1680.965	0.800	33.479	543.323

Table 1: Experiment results

Mainly due to Pr_{miss} being conditioned on the fact that there must be a segmentation/post there. Posts come in bursts, visits are fairly periodic, and intervals between visits are larger than post bursts. More posts than visits in places with posts, hence higher Pr_{miss}

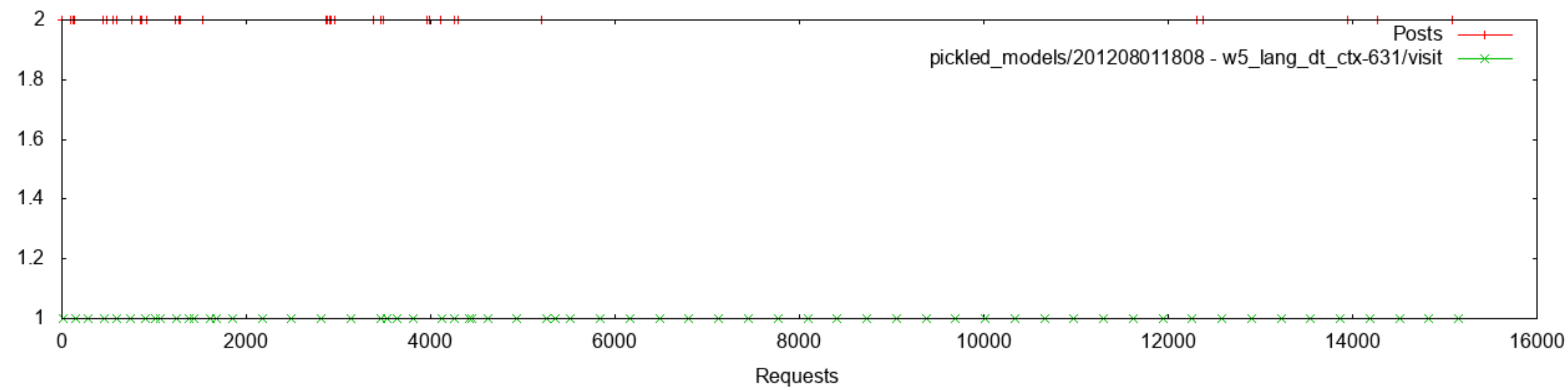

Figure 1: Visitation chart for a model using the $w = 5, \mathbf{t}_{\Delta}, \mathbf{t}_{\text{ctx}}, \mathbf{w}$ feature set. Invalid Predictions = 0.758, $Pr_{error} = 0.485$, T-score = 119.612, Posts = 41, Visits = 62

High no. of invalid predictions.