# A Deep Semantic Search Method for Random Tweets

Isa Inuwa-Dutse, Mark Liptrott and Ioannis Korkontzelos

*Department of Computer Science, Edge Hill University, UK*

**Abstract**

Contemporary social media platforms enable users to act as both producers and consumers of content, leading to the generation of enormous amounts of data. While this ability is empowering, it is also posing many challenges concerning efficient searches for relevant information. Many search approaches have been proposed in the literature. However, searching for information on Twitter is particularly challenging due to both the inconsistency in writing styles and the high generation rate of spurious and duplicate content. The quest for instant and efficient data processing to retrieve relevant information renders many existing techniques ineffective when applied to Twitter.

We present a multilevel approach based on state-of-the-art deep learning methods and a novel scalable windowing approach for pairwise-similarity search (SWAPS) to improve search efficiency. SWAPS optimises searches using a strategic balancing criterion to assess the trade-off between accuracy and search speed, thereby circumnavigating sequential search problems. Moreover, we propose a deep search strategy that establishes a relationship between the status of a tweet and its longevity measured in terms of engagement lifespan since posting. Deep search utilises a convolutional neural network for textual *n-grams* features extraction and meta-features from the tweet to train a fully connected network on a vast number of tweets. This approach differs from existing ones by recognising the relationship between the status of a tweet and its engagement lifespan to ensure a better understanding of the compositional semantics in tweets. The results highlight interesting symmetrical properties with respect to similarity distribution and duration. We evaluate our approach on various benchmark datasets and demonstrate the efficacy and applicability of the method. Problems of event detection, clustering and ads, among others, can utilise this approach to detect items of interest effectively.

*Keywords:* deep learning, semantic search, tweets, Twitter, information

## 1. Introduction

Since the inception of the world wide web, the mode of interaction between the media and the public has shifted from the traditional 2-step flow [18] to multi-flow [42] where users act as both producers and consumers of information. This culminated in a period of a rapid data growth that is posing computational challenges to tasks[1] where pairwise similarity is central. Various measures[2] have been taken to improve interaction in terms of navigation and information search. The continuous increase in online content often poses challenges to interact effectively with online sites. Some measures to address the challenge range from the positioning of URLs at suitable locations to bookmarking information resources based on semantic similarity. For instance, the work of Dourish and Chalmers [8] examined the underlying semantic relationship between information-bearing objects in spatial models of navigation. Heymann et al. [14] leveraged the availability of user-generated data, e.g. tags, bookmarks or any form of rich annotation in the web that provides useful data, to improve online search and navigation. Enhancement techniques based on heuristics and careful engineering of features have also been considered in Aggarwal and Subbian [1]. Information about some implicit factors such as interests, culture or geolocation as outlined in [28], have been shown to improve online information searches [13].

*Searches on Twitter:* By enabling users to annotate contents, e.g. *#hashtag*, search for information has been greatly simplified on Twitter. Users can perform a basic search using *usernames, hashtags*, *trending topics* or any meaningful *keywords*. While these annotations have been shown to improve searches [46], the high production rate of content from *influential users* often eclipse less popular content [15]. As the volume of data in the social media ecosystem increases, a variety of options are open for exploration. This study posits that relevant information can be searched efficiently as a function of time. We propose a multi-level search method based on deep learning and a novel *scalable windowing approach for pairwise-similarity search (SWAPS).*

[1]For example, in topic detection and tracking (TDT), clustering, event detection or database search.

[2]For example, the early SMART project offers a test-bed to implement and evaluate IR tasks [36].

To illustrate our proposed approach, consider Figure 1 that shows the result of a sequential pairwise-similarity search between an anchor tweet $t_a$ and other tweets $t_i$ in a window $w$ of size $z$. An *anchor tweet* $t_a$ is the focal point of computing pairwise similarity with *other tweets* in a *window*. Let $t_a$ and $t_i$ be two tweets posted at times $q$ and $b$, respectively. We aim at estimating the time interval $b - q$ after time $q$, until a similar tweet $t_i$ to $t_a$ is found, given that this relative time difference is found within window $k$ of size $z$ ($w_k^z$). See Table 1 and Section 3 for full notations and definitions in the study. Our goal is to efficiently identify tweets similar to the anchor, without searching sequentially. The proposed *SWAPS* is based on the premise that *if we could predict the high-density area of the most similar tweets in a window, then we can effectively find a group of similar tweets to any tweet without searching sequentially*. Firstly, we apply probabilistic reasoning to quantify the degree of uncertainty in a set of tweets with respect to the approximate similarity to any tweet within the same collection window and the time spent for the search. The task proceeds by estimating the distribution of similarity and relevant statistical quantities in random tweets to design an effective search method.

We then apply a deep learning technique to predict the *engagement lifespan* of a tweet as a function of its status. This establishes the relationship between the status of a tweet and its engagement lifespan, which is defined as the duration of wider engagement with the post after being posted. Deep learning methods are powerful tools to automatically extract *lexical-level* and *sentence-level* features without resorting to handcrafted rules. To understand the relationship between the status of a tweet and its engagement lifespan, lexical features have been extracted using a *convolutional neural networks (convnets)* and used in training a fully-connected neural network using over 60 million pairs of tweets.

### *1.1. Contributions*

The increasing high generation rate of online content, which makes searching for relevant items difficult, is the motivating factor behind this study. To enhance searching, we contribute the following:

- We statistically analysed the distribution of similar items across 5 benchmark datasets and two collected for this study (see Table 2). Accordingly, we conducted rigorous statistical tests and interpretations with respect to population parameters, i.e. the *confidence interval at*

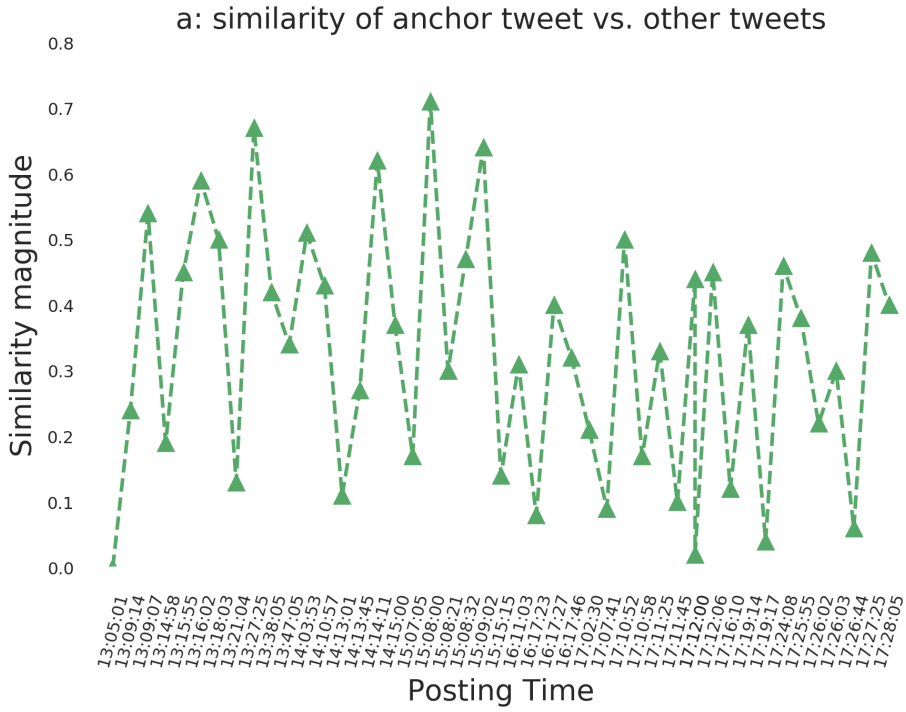


Figure 1: A time-similarity graph showing variations in terms of similarity magnitude in a window. Peaks denote tweets similar to the anchor tweet where the similarity decays overtime

*which to expect adequate similarity in a finite window, the sample mean and variance* to provide useful practical insights. This enabled us to identify the prior parameters applicable for estimating similarity in related tasks involving pairwise-similarity, e.g. clustering.

- We present a novel search algorithm (SWAPS) that balances the trade-off between search speed and the number of relevant discovered items. SWAPS efficiently returns relevant items with minimal loss of accuracy in comparison with a sequential approach.

- We provide a deep learning strategy that leverages the powerful *convnets* framework to extract relevant features to predict the engagement lifespan of a tweet. The strategy optimises search problems by highlighting the meaning and the symmetrical property in terms of how similar tweets tend to rally around tweets of high status. The strat-

egy could be useful for search and advertisement scheduling since it estimates when high user engagement is expected.

- The developed datasets will be made freely available.

This paper is structured as follows. Section 2 reviews related work and the subsequent Section 3.1 describes the research data. Section 3 introduces the proposed approach and experiments. Section 4 presents the discussion. Section 5 concludes the study and proposes some future work. Table 1 shows a summary of the notations utilised in the paper.

Table 1: Notations and descriptions

| Notation | Description |
|---|---|
| $t_a$ | an anchor tweet |
| $t_i$ | other tweets being compared with $t_a$ |
| $w_k^z$ | a finite collection of z tweets in window k |
| $t_{circle}^{a_i}$ | set of tweets with high similarity with the anchor tweet |
| $\chi$ and $\gamma$ | sets of training examples and target labels respectively |
| $\{x_i, y_i\}_{i=0}^n \in \mathbb{R}$ | a training instance |

## 2. Related Work

This section reviews research related to *searches for relevant items*, *search enhancement* and *deep learning methods in Natural Language Processing*.

*Relevancy search.* Effective searches for similar items have been of major concern for a long time. The early work of Agrawal et al. [2] proposed an indexing strategy using the *Discrete Fourier Transformation (DFT)* that maps sequences from the time domain to the frequency domain and computes similarity using Euclidean distance. Rafiei and Mendelzon [35] extends the approach in [2] to identify similar queries based on sequence matching. These techniques rely on sequence matching to evaluate similarity, which is limited in capturing rich semantic relationships. Vlachos et al. [41] applied the DFT analysis to discover similar queries by comparing query signals from search engine logs. Peng et al. [33] also applied DFT to analyse word trajectories in both time and frequency domains. Words exhibiting signal patterns along

the trajectories are considered relevant and the higher the signal peak, the more relevant the item. The diversity in tweets, due to the non-standard style of the text, limits the applicability of this approach on Twitter. The varying degree of growth and intensity exhibited by social media content has been investigated in [43], [19] and [26] to reveal underlying mechanisms. In Twitter, bursty patterns have been shown to follow basic statistical distributions, such as the *power law*, and to be mostly triggered by influential users, making other tweets not subscribed to such trends go unnoticed.

*Search enhancement.* The growing volume of online content challenges effective filtering for relevant data. This has prompted various strategies to enhance the process. Early mitigation strategies include bookmarking or collaborative social tagging [31], [14], [46] and optimisation strategies either based on heuristic or careful engineering of features [13], [1]. For example, Lagnier et al. [21] investigates how *information diffuses* within communities, based on interaction dynamism, users' willingness to transmit and the generated content, to study diffusion patterns and ultimately improve online searches.

With respect to design principles and operation, closely related work to ours can be found in [6] and [38]. Chen et al. [6] proposed a tweet indexing method (TI) for real-time search based on keywords. With the growing complexity of social stream and synonymous terms, indexing based on the exact match will have limited coverage since synonymous terms will be overlooked. Our approach does not require indexing tweet, but its aggregation based on semantic features learned overtime. Sundaram et al. [38] proposed a *Locality-Sensitivity Hashing (LSH)* approach that identifies duplicate or near-duplicate documents. Hashing algorithms are sensitive to variations in input where synonymous words may end up in different regions in the hashtable. The LSH technique is a useful strategy to avoid sparsity problems; however, similar to Chen et al. [6], the LSH does not account for synonymous terms in the documents being compared. Our study utilises convnets which serves to account for synonymous terms and variations. Convnets contextually aggregate words with rich semantic similarity with closer distance or proximate in the vector space, hence more comprehensive.

*Deep learning methods.* The increasing volume of social media data requires proportionate handling tools, and prior research works have identified deep learning models as most useful. Recently those models have revolutionised

many research areas from basic computations to complex computer vision tasks, such as real object recognition in images or videos. Since the pioneering work of Kim [20] on *convolutional neural networks (CNN)* for text classification, there has been a surge of implementations and useful best practice for various *NLP* tasks [45]. Sutskever et al. [39] applied deep learning techniques, in particular, Long short-term memory (LSTM) units, for textual sequence mapping, applicable to automatic translation tasks. Deep learning has been successfully applied to numerous extraction tasks [7], [44], [24], due to its capability to automatically extract lexical-level and sentence-level features without resorting to handcrafted methods or cumbersome traditional NLP tools. Mencia and Fürnkranz [29] and Bhatia et al. [4] applied the traditional *multilayer perceptron* and deep learning for multi-label classification, respectively. Motivated by the success of *deep learning* in related areas, this study leverages it to efficiently search for relevant information on Twitter, thereby contributing to search enhancement.

## 3. Proposed Approach

This section describes our multi-level approach based on the proposed *SWAPS algorithm* to speed-up searching and *deep learning* to predict *engagement lifespan*. We begin by quantifying the uncertainties associated with the problem using a probabilistic inference toolkit[3], describing SWAPS and finally introducing the deep learning strategy.

*Notations and definitions.* For the prediction task, $\chi$ and $\gamma$ denote sets of training examples and target labels respectively and $\{x_i, y_i\}_{i=0}^{n} \in \mathbb{R}$ denotes a training instance. The input, $\chi$, consists of both main and meta features (see Table 2). For any anchor tweet $t_a$, its *circle* consists of most similar tweets to it denoted by $t^a_{circle}$. For a given window, each anchor tweet is represented as a list of tuples containing the similarity score ($\phi$) between the *anchor tweet* and any other tweet $t_i$, and the *relative posting time* $p = p_j - p_a$ in seconds[4]. Thus, for each anchor tweet

$$t^{a_i}_{circle} = [\phi((t_{a_i}, t_j), p)]^{n}_{j=i+1}$$

[3]We utilise the *PYMC3* probabilistic programming toolkit developed in Python [37].

[4]$p = p_j - p_a$ defines the time difference between the anchor tweet and a closely related tweet only, $j$. Similarly, $p = p_i - p_a$ defines the time difference between the anchor tweet and any other tweet, $i$.

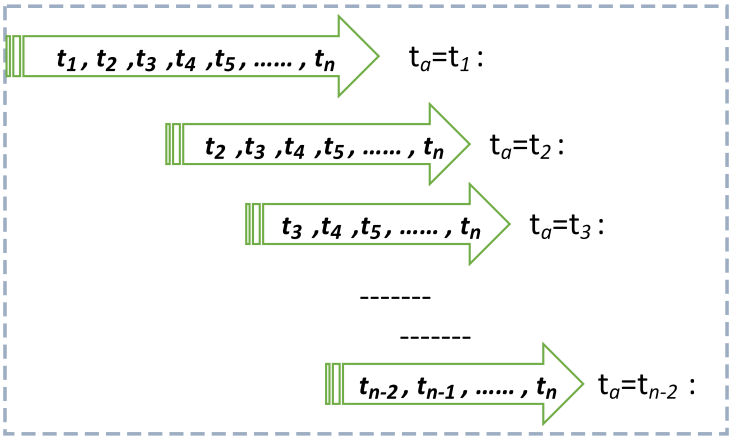


Figure 2: An example of how each tweet in a finite collection of tweets compares with others. Each tweet is a potential anchor, and for each designated anchor, $t^{a_i}$, in the window, the set of tweets whose similarity is higher than a threshold $\tau$ constitutes the anchor's circle given by $t^{a_i}_{circle}$. Note that $j \in t^{a_i}_{circle}$ refers to a tweet with a high degree of similarity with the anchor tweet which distinguishes it from other tweets $t_i$ that could be similar or dissimilar to the anchor.

where $n = |\phi(t_{a_i}, t_j) \geq \tau|$ and $\tau$ is a predefined threshold[5].

*Definition.* Similar tweets refer to any pair of tweets $(t_a, t_i)$, whose similarity magnitude $\phi$ is greater than a predefined threshold $\tau$. We denote $\phi(t_a, t_i) \geq \tau$ as a random variable that defines a similarity between the anchor tweet $t_a$ and any other tweet $t_i$, otherwise dissimilarity (i.e. $\phi(t_a, t_i) < \tau$).

Computing pairwise similarity on large document collections is a task common to a variety of problems. Similarity metrics are broadly categorise as *sequence matching* and *linear (word-embedding)*. Sequence matching computes similarity by matching co-occurrence of lexical sequences in documents using metrics such as *Cosine* and *Dice* [2]. These metrics suffer a setback if apply to tweets due to the sparsity of co-occurring terms [23]. Similarities based on *word-embedding* can reveal the semantic similarities since it does not rely on matching co-concurrence but the contextual meaning of terms. Common examples are *doc2vec* and *word2vec* [30]. Our approach computes similarity based on *word embedding*.

[5]For all experiments in this paper, $\tau = 0.5$, i.e. two tweets are considered similar (1) if $\phi \geq 0.5$, otherwise dissimilar (0).

### *3.1. Dataset Description*

In this section, we describe the datasets and the corresponding preprocessing technique given in Section 3.4.1. We utilise two categories of data: *Collected data* (collected mainly for the study) and *Public data* (available from public data repositories). Majority of the datasets consist of collections of short messages (known as tweets) obtained from Twitter.

#### *3.1.1. Collected Datasets*

This data category is mainly collected for the study and consist of: *subject-based tweets (SBT)* and *diverse tweets (DVT)* datasets. Both the *SBT* and *DVT* consist of tweets collected from Twitter using a collection crawler that returns relevant information based on *keywords*. Keywords play a crucial role in retrieving specific documents from a large corpora [25]. Our collection approach is based on *ad-hoc retrieval* method, which involves the use of descriptive keywords to search for relevant documents [27]. The *SBT* consists of a collection of tweets posted during the height of the *EU refugee crisis (2016/2017)*. Noting the bias that may arise due to the seemingly black box sampling strategy by Twitter in returning queried documents [40], we utilise diverse keywords covering many aspects of the subject. Sample collection keywords include *refugee, migrants, refugee crisis, EU refugees, refugees & (refugee/migrants); migrants & (refugees/migrants); crisis & (refugees/migrants)*. The *DVT* consists of a random collection of tweets spanning diverse topics of discussion on Twitter based on [3]. This is to mitigate similarity bias likely to be caused by focusing on the specific discussion topic and to maximise the diversity and randomness in the data.

#### *3.1.2. Public Datasets*

In addition to the data purposely collected for the study, we use the following datasets, which can be downloaded from public data repositories.

*Review tweets* These are collections of reviews posted by users on Twitter. *Review tweets1* [11] consists of reviews about drugs and *Review tweets2* [17] contains a collection of tweets about health-related issues from major health news agencies. *Review tweets3* [5] consist of customers' reviews about services offered by hotels. It is expected these datasets will have a high degree of similarities, which will be useful for evaluation.

*Eur-Lex Dataset [29]* This is the only non-tweet dataset in the study. We use the *Eur-lex dataset* which does not incorporate temporal information to demonstrate the operation of *SWAPS* beyond tweets.

Table 2: Datasets and features (main and meta features comprising the tweet signature)

| | Group | Pairwise Size | Unique | Description |
|---|---|---|---|---|
| Datasets | Diverse tweets (DVT) | 35m | 300000K | consists of random tweets collected using diverse keywords covering many domains (based on [3]) to introduce a high level of randomness and improve the universality of the dataset. |
| | Subject-based tweets (SBT) | 45m | 300000K | consists of tweets collected in 2016/2017 related to EU refugee crisis |
| | Review tweets1 | 3107 | 602400 | data about patients' reviews on specific drugs and related conditions |
| | Review tweets2 | 334140 | 17413 | contains tweets about health news from major health news agencies |
| | Review tweets3 | 400 | 9336 | online and offline collections of customers review about hotel service |
| | Social Circles | 27549 | 7059 | collection of tweets from social circles in Twitter |
| | Eur-Lex | 12353646 | 62311 | data about EU legal documents |
| Features | main features | – | varies | consist of various n-gram features (unigram, bigram, trigram, forth-gram and fifth-gram) extracted from each tweet using deep learning convnets |
| | meta features | – | 6 | extracted from the meta-data of a tweet or the user that posted it. Features include similarity score, relative posting time, period (e.g. morning, afternoon), number of followers, tweet's favourite count, number of friends |

*Social Circles Dataset [22]* This dataset consists of 81,306 users crawled from Twitter. Based on the IDs of the users, we retrieved their tweets and other relevant information for the study. Because this dataset is from users with affiliations to specific online communities [22], we expect a higher degree of similarity in their tweets. We extracted 7059 unique users for this experiment.

### *3.2. Uncertainty quantification*

Figure 3 shows a hypothetical finite window $k$ of size $z$ ($w_k^z$) depicting how an anchor tweet $t_a$ compares with all other tweets in the window. Conventionally, $t_a$ is sequentially compared with every other tweet $t_i$ in the win-

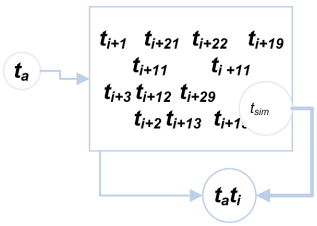


Figure 3: A hypothetical finite window of random tweets depicting a random anchor tweet, $t_a$, posted at time $p$, its $m$ similar tweets and the time distance $d$ between a similar tweet $t_{sim}$ and the anchor tweet $t_a$

dow, and this approach certainly impedes the efficiency of the process if the search space is vast. Our preliminary analysis, shown in Figure 1, suggests that a tweet tends to have a set of $m$ most similar tweets known as *circle*, distributed within a finite window. The goal is to compute the probability distribution of *similar vs. dissimilar* tweets in relation to a random anchor tweet in any given window. Accordingly, we conduct rigorous statistical tests and interpretations with respect to the population parameter (i.e. the *confidence interval for true similarity distribution in a finitely sized window*), the *confidence interval*, *sample mean* and*variance* to gain useful insights applicable in practice. We experiment using a diverse collection of tweets and various window sizes. In other words, we take a bootstrap sample from the corpus population, that is useful in measuring the variability of the similarity distribution and their temporal behaviour in the windows.

*3.2.1. Distribution of similar tweets in a window*

The *circle* size $m$, $(m = |t^a_{circle} \in w^z_k|)$, of an anchor tweet is considered as a random variable $\theta$ such that $\phi(t_a, t_i) \in [0, 1] > \tau; i = 1, 2, 3, ...., z$. In line with related studies [10], [23], [12], we apply probabilistic generative models to estimate $\theta$ in each window. We begin with Bernoulli:

$$\phi(t_a, t_i)\ Bern(\theta) = Bern(1, \theta)$$

such that

$$p(\phi(t_a, t_i) = 1|\theta) = \theta$$

and

$$p(\phi(t_a, t_i) = 0|\theta) = 1 - \theta; 0 \leq \theta < 1$$

The respective *mean* and *variance* are given by $E[\phi(t_a, t_i)] = \theta$ and $var[\phi(t_a, t_i)] = \theta(1 - \theta)$. For a finite window, the estimation follows a Binomial distribution

where $m$ denotes $\phi(t_a, t_i) = 1, \forall i \in m$ and the sum of possible ways to obtain $m$ given by:

$$Bin(m|z, \theta) = \binom{z}{m} \theta^m (1-\theta)^{z-m}$$

For each window, we repeatedly compute the number of ways $\binom{z}{m}$ to select similar tweets (if any) to the anchor tweet on different samples by varying the window sizes. This approach enables us to quantify the required number of trials in a window, the window size required for finding enough similar tweets and the associated uncertainties regarding the variability of $t^a_{circle} \in w^z_k$. Figure 4 shows results from trials that utilised various window sizes. Using a relatively small window size of 200 tweets shows high instability and many dissimilar tweets. Increasing the window size to about 500 tweets provides more stability and increases the number of similar tweets. The distribution remains virtually unchanged with a window size of 1500 and 2500 random samples.

The information in Figure 5 allows the computation of statistical quantities about the data such as the *sample mean*, the *median*, the the *highest posterior density (HPD)* and the the *region of practical equivalence (ROPE)*. In the Figure, *HPD* and *ROPE* are represented as a long black bar and red bar respectively. The *HPD* quantifies the probability that there is a 95% dissimilarity between the expected and the actual data distribution[6]. This is crucial in deciding whether to increase the window size or not.

Figure 6 shows a pair-plot of the mean similarity and mean duration as a function of various window sizes. Many similar tweets in the range of 500-800 can be observed and the duration or relative posting time spans to -15, which suggests that there exist tweets similar to the anchor tweet before its posting time.

### *3.3. SWAPS Algorithm*

Informed by the quantification of uncertainty and the related statistical quantities, we present the *Scalable Windowing Approach for Pairwise-similarity Search (SWAPS)* algorithm. *SWAPS* utilises the expected *mean*, $E[\phi](mean_{base})$, and the *variance*, $var_{base}$, as *baseline parameters* to regulate

[6]The computation can be conditioned on the time of collection and the popularity of the content since influential users on Twitter attract more attention and drive trending topics

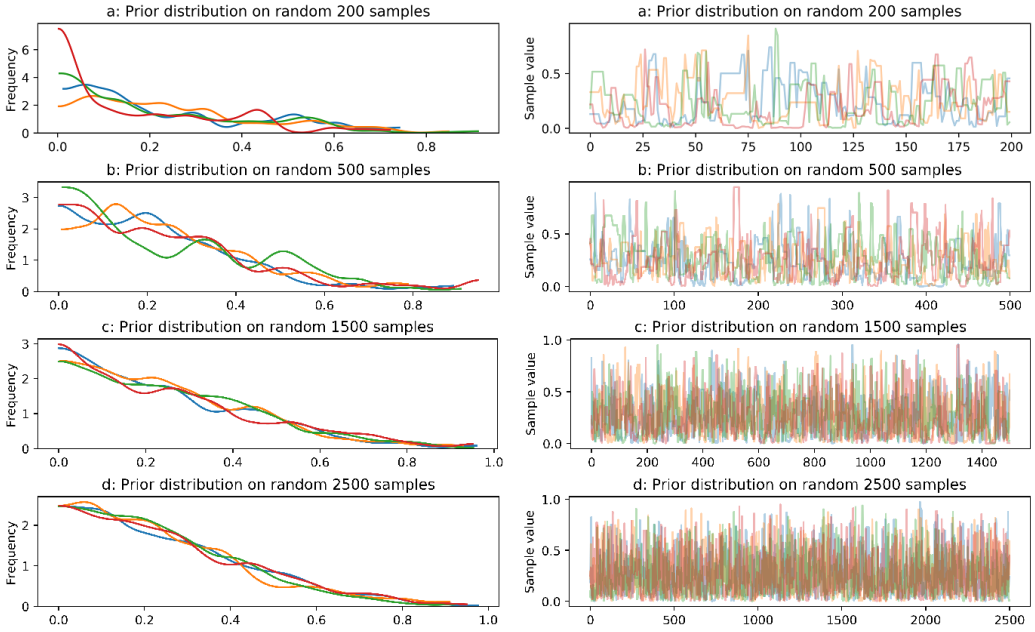


Figure 4: The *trace* or results from random samples drawn from the *posterior distribution* based on the *prior*. The different colours in the line denote similarity values of samples; $x$ and $y$ axes respectively denote the similarity value and frequency in the window. The corresponding sub-figures in the right column report the similarity magnitudes and degree of stability in the samples as a function of window size. We can observe a drop in the perturbations as the window size is increased: (a) a small window size, 200, shows a high level of instability and low similarity (b) the instability is still evident with a window size of 500 (c) a window size of 1500 shows moderate stability and increased similarity (d) finally, a window size of 2500 shows no major improvement over (c). The distribution remains virtually unchanged with a window size of 1500 random samples

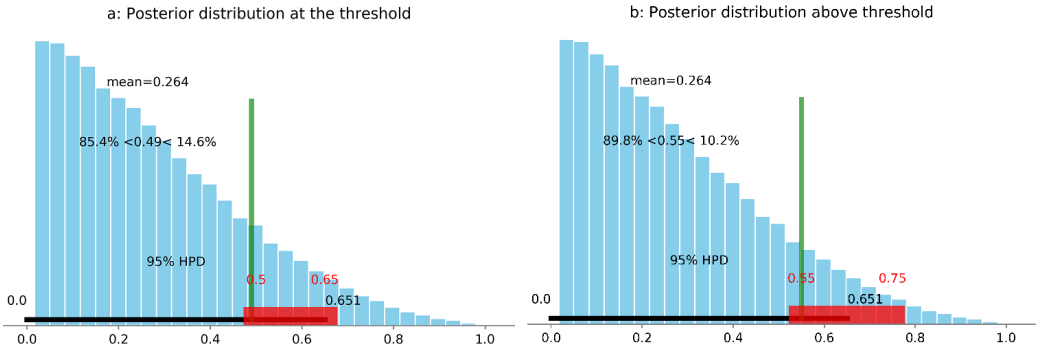


Figure 5: The posterior distribution and the quantification of uncertainties, based on the observed data. Relevant statistical quantities about the data such as *sample mean, median, the highest posterior density (HPD) and the region of practical equivalence (ROPE)* can be defined. *HPD* quantifies the belief that on the distribution corresponding to our expectation and the observed data, 95% will be dissimilar. The ROPE is useful in deciding whether to keep increasing the window size or not by using values within the desired threshold. For instance, The red *ROPE* along the *black HPD* bar in (a) at the threshold value and (b) above the threshold value corresponds to a tunable region where various values can be evaluated, e.g. $5.5 - 6.5$

its operation. These quantities can be used to evaluate and inform assumptions, such as the actual mean similarity in an interval $[p - q] \in w$ express as $\mu_{sim}([\phi]_p^q)$. This quantity should be at least equal or greater than the $mean_{base}$ such that

$$\mu_{sim}([\phi]_p^q) - E[\phi] \geq 0$$

The mean value is useful in taking longer search steps, and the variance informs how the distribution changes at shorter intervals. A significant deviation in these quantities heralds a change in the similarity distribution. For instance, the undulations in Figure 1 are related to changes in the mean and variance and are utilised by the algorithm to decide when and how to regulate the search process. Consequently, a control mechanism consisting of catalysing factor $c$ (or *c-factor*) and *jump index* (or *j-index* $\varphi(t_j)$) is proposed to effectively guide the process. The *c-factor* is related reciprocally with the mean similarity. The *j-index* accepts the *c-factor*, $\mu_{sim}([\phi]_p^q)$, $|w_k^z|$ and the current position of the anchor tweet to compute the next arbitrary starting point $j$. These quantities are related as follows:

$$\mu_{sim}([\phi]_p^q) = \frac{1}{|w_k^z|} \sum_{p}^{q} \phi(t_i, t_j) \geq \tau \qquad t_i, t_j \in p - q$$

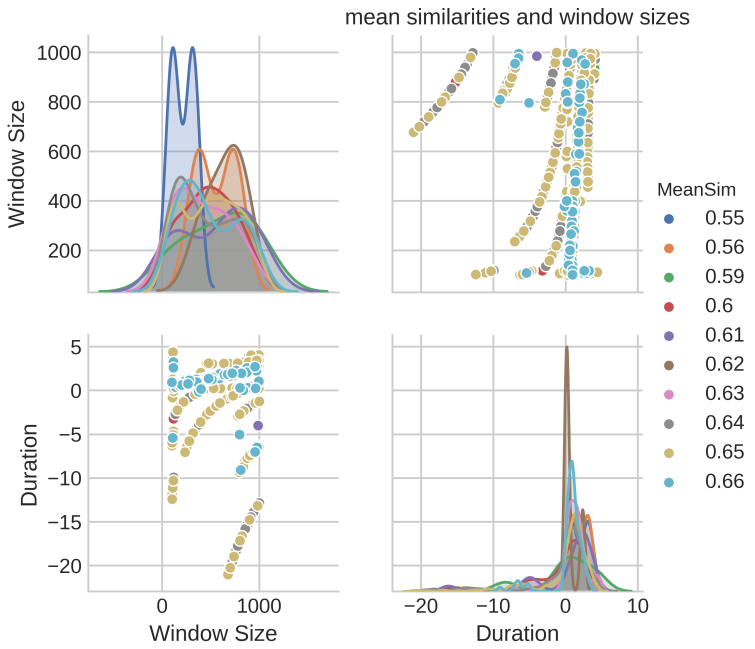


Figure 6: A pairplot to explore the relationship between mean score, window size and relative posting time as a grid of axes. Each variable in the plot is shared in the *y-axis* across a single row and the *x-axis* is the same along the column. A reasonable amount of similar pairs are obtained using a window size of about 400-500 tweets. More similar tweets can be obtained by increasing the window size to the region of 800-1000 tweets.

$$c = \frac{1}{\mu_{sim}([\phi]_p^q)}$$

$$\varphi(t_j) = c + \frac{|w_k^z|}{p - q \times \mu_{sim}([\phi]_p^q)} \qquad \varphi(t_j) < |w_k^z|$$

The *j-index* always returns an integer value less than the window size, which corresponds to the position where searching should continue. The *search limit (l)*, a user define fractional value, defines the point to invoke *SWAPS* after $n$ sequential search steps. Our implementation uses $l = 1/4$, i.e. 1/4 of the space has been searched before SWAPS is invoked. An interesting property of both $c$ and $\varphi(t_j)$ is their diminishing behaviour, as illustrated in Figure 7, as the mean similarity increases over time. SWAPS can accommodate any standard similarity metric, such as *Cosine similarity*, or a

custom metric to suit the application requirements. For instance, the performance of the Cosine similarity, which is based on terms co-occurrence, can be enhanced by incorporating the length of terms, based on the observation that terms that consist of many characters has been shown to be informative [34]. However, a more powerful approach is to compute similarity based on word embedding vectors [30]. Our analysis results are based on computing *Cosine similarity* (Section 3) between the embedding vectors of a pair of tweets.

**Algorithm 1** *SWAPS*: Given a set of timestamped tweets $t_i, \ldots, t_{z-1} \in w_k^z$) posted at time $p \in [p_k, p_m]$ from corpus $D$:

**Initialisation:** *anchor tweet* $t_a$, *buddy tweet* $t_i$, *baselines* $(mean_{base}, var_{base})$, *search limit* $l$
**while** $t_i index < z - 1$ **do**
    $\forall t_a, t_i \in w_k^z$ compute $\phi(t_a, t_i); \mu_{sim}; var_{sim}$
    **if** $t_a index \geq l \times |w_k^z|$ and $\mu_{sim} < mean_{base}$ **then**
        compute $c$ and $\varphi(t_j)$
        *update anchor:* $t_a index \leftarrow \varphi(t_j)$
        *update buddy:* $t_i index \leftarrow \varphi(t_j) + 1$
    **else**
        $t_a \leftarrow t_a$
        $mean_{base} \leftarrow \mu_{sim}$;  if $mean_{base} < \mu_{sim}$
        $var_{base} \leftarrow var_{sim}$ ;  if $var_{base} < var_{sim}$
        continue
    **end if**
    $t_a \leftarrow t_{a+1}$
    $t_i \leftarrow t_{i+1}$
**end while**

*3.3.1. SWAPS Complexity*

Both sequential search and SWAPS are iterative algorithms, and operation-wise they are similar since activities such as looping over items are common in both, but the items to be compared are different. For a finite window $w_k^z$, the total number of comparisons to be made by the sequential process is $\sum_{i=1}^{z-2} z - i$ and by SWAPS approximately $\frac{1}{3} \sum_{i=1}^{z-2} z - i; i \neq z$. Both methods leverage the symmetry in dot multiplication for computations. In SWAPS, there is approximately 1/3 chance of invoking a control mechanism. With a window size of just 350 tweets, there are 30625 total comparisons to be made

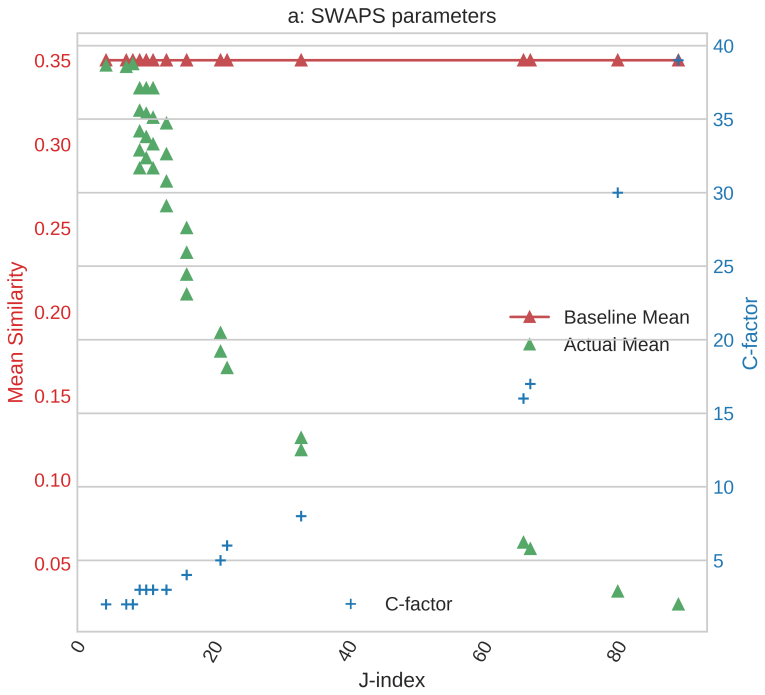


Figure 7: The behaviour of SWAPS parameters in regulating search speed: as the mean similarity improves, the effect of the *c-factor* (labelled as + at the bottom middle) is diminishing, hence shorter or no *j-index* values apply.

using sequential searches and roughly 10208 comparisons using SWAPS. Table 3 summarises the execution time, measured in seconds, for both methods on various datasets. For the sequential approach in window $w_k^z$, the increment is linear and the first $m$ anchors will cause $z-1, z-2, z-3, \ldots, z-m$ comparisons, respectively, with complexity of $O(z^2)$. The execution of the outer loop in SWAPS, shown in Algorithm 1, is dependent upon the control mechanism that decides the next starting position. Considering a minimum jump, based on the mean similarity of 0.2 after 170 pairs have been searched, *c-factor* evaluates to 5 and the corresponding *j-index* to 15. Proceeding at a steady pace for $m$ iterations, the complexity is bounded by $O(z(logz))$.

### *3.4. Status of a tweet and engagement lifespan*

The motivation behind the use of deep learning is to strategically optimise searches by exploring the idea that relevant information can be efficiently searched as a function of the *engagement lifespan* of tweets. To learn the

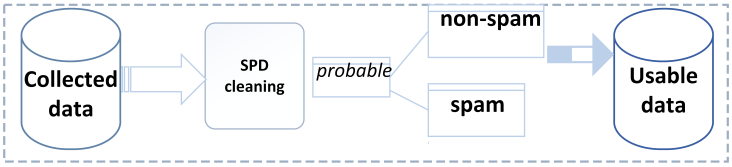


Figure 8: Data cleaning pipeline. The *SPD cleaning* step involves the use of a prediction model trained on numerous features related to tweets.

association between the *status* of a tweet and its *engagement lifespan*, over 60 million tweets have been represented based on their meta-features and used to train a regression model. The goal of the regression model is to predict *time interval* at which a given anchor tweet will attract significant attraction. The predicted time is not absolute but an approximate time range, in which relevant tweets to the anchor tweet are expected to be posted.

*3.4.1. Data Cleaning and Preprocessing*

Online social media attract all sort of information from diverse users. In terms of cleaning, social media data is particularly challenging to process due to the prevalence of a personalised form of writing and lack of structure emanating from a lack of standard writing styles [32]. Tweets are generally noisy and constitute a substantial proportion of irrelevant or spam content which could undermine analysis result. As an initial preprocessing task, we utilise a spam filtering technique (*SPD*) proposed in [16] to get rid of irrelevant content from the data. The *SPD* approach makes it possible to incorporate detection mechanism in the data collection pipeline or apply the technique to an existing data to remove tweets with a high probability of being spam. To enable the use of SPD, we collect all the features such as *network features* and *textual features* required for the SPD detection, which returns the likelihood of spam or non-spam. Figure 8 shows the data cleaning pipeline. The filtered tweets are then normalised by converting to lower case and removing stopwords to obtain *shingles*. Shingles are the set of attributes for similarity comparisons and are obtained after the removal of URLs, #hashtags, @mentions[7].

[7]These sequences were removed so that the approach generalises well on text other than tweets, such as the Eur-Lex data (see Table 2)

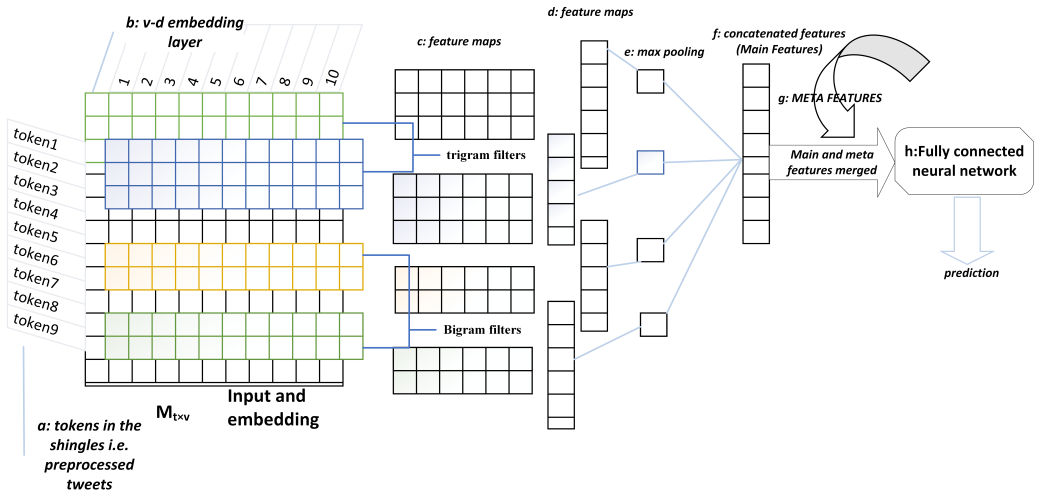


Figure 9: A convnet and a fully connected neural network consisting of numerous dense layers. The framework uses main and meta features to learn the relationship between the status of a tweet and its engagement lifespan. Each *channel* in the architecture consists of (a) the length of the *input sequence* (b) an *embedding layer* (c) an *1-d convnet layer* with 32 *filters* and a *kernel size* (equivalent to the n-gram) (e) a *max pooling layer* to select best feature from (d) the *feature map* and finally (f) integrates the *output* which combines with *meta features* to train (h) a fully connected neural network.

### *3.4.2. Feature Extraction*

For training and evaluation purposes, Figure 9 shows the *feature extraction* and the *training pipeline* utilised in the study. Two sets of features have been used: *main features*, i.e. features extracted from raw tweets using 1-d multi-channel convolutional neural network (*CNN* or *convnet* based on Kim [20]), and *meta features*, which are fed to the fully connected neural network segment in the figure. *Convnet* automatically extracts relevant features in a tweet as *n-grams* at various lexical-levels. Table 2 provides additional details about the features.

The filter in each channel of Figure 9 is initialised with the embedding of the term as a weight vector[8]. We adjust the region size of filters to have the same width as the dimensionality of the word vector, to preserve the inherent sequential structure in the data [45]. The fully connected neural network (FCN), i.e. segment (h) in Figure 9, accepts the main features, con-

[8]The embedding matrix $E$ is trained on the SBT and DVT datasets.

sisting of the integrated best features from each channel defining a high-level representation of a tweet, and the corresponding meta-features of the tweet for training. As far as the variations in the data scale, with respect to meta-features, are concerned, each feature used for training the deep learning model is proportionally scaled by subtracting the mean and dividing by the standard deviation. Finally, the data is proportionally split into train and test sets.

*3.4.3. Prediction*

We conducted a series of experiments on various datasets, shown in Table 2, to learn the behaviour of as many different tweets as possible at various times.

The training objective is to minimise crucial loss functions: *mean-squared error (MSE)*, and *mean absolute error (MAE)*. These are useful metrics to assess the efficacy of the model. Figure 10, which utilises the *MSE loss function*, shows the experimentation results using the $SBT$ and $DVT$ datasets.

*MSE.* This is a widely used loss function in regression problems which is expressed as the mean of sum of the squared distances between the target ($y$) and the predicted ($\hat{y}$) values:

$$mse = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The aim is to minimise the distance or error between the true value and the predicted value.

*MAE.* In addition to the MSE which takes into account the direction of the errors, we apply the MAE which evaluates the mean magnitude of the errors in the prediction task. It is based on the absolute difference between the target ($y$) and the predicted ($\hat{y}$) values:

$$mae = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$$

In Figure 10(d), the target values appear as a straight line due to proximity in posting time between the tweets which were collected from the diverse topic of discussion. With an average 100m daily users contributing about

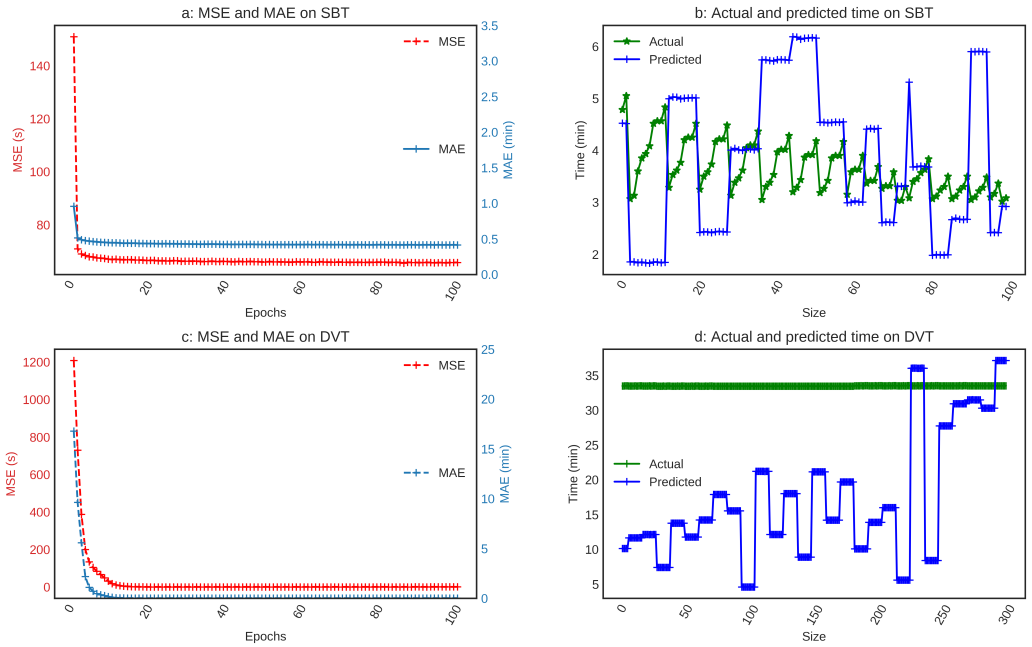


Figure 10: Evaluation results on SBT and DVT datasets. Sub-figures (a) and (c) depict the MSE and MAE, and sub-figures (b) and (d) compare the actual target to predicted targets. There is a shorter time interval in the DVT dataset, which can be explained due to the random collection of topics discussed simultaneously in the data and the predictions are mostly behind the target. See Table A.4 for some examples. There is a longer duration in the SBT, which is generated using a small set of specific keywords to filter relevant content but performs better.

500m content[9], the amount of tweets is enormous. Within a second or two, thousands of tweets are being produced, and because the collection keywords span various discussion topics on Twitter, many unrelated tweets are produced. We reduce the size, as shown in the appendix (Figure B.15) to make the pattern more visible.

The diverse datasets, collected using keywords spanning broad subjects such as *sports, entertainment, politics, education, news*, consist of a multitude of disparate tweets from these broad categories posted within a short period with close proximity. Thus, the probability of picking a tweet with high similarity with other tweets in the *DVT* is evidently low. The diverse dataset is

[9]See https://www.omnicoreagency.com/twitter-statistics/

actually not the ideal practical use-case since real information search often starts with some high-level keywords specific to the search topic. The rationale of using the *DVT* is to compare with the ideal use-case that employs specific searching criteria using the *SBT*, which shows a better performance with potentials of improvement.

*3.4.4. Evaluation*

We conducted three forms of analysis: (1) several quantitative evaluations on various test datasets (2) a comparative analysis between SWAPS and sequential search, and (3) an evaluation on independent benchmark datasets.

In addition to the SBT and DVT, we utilised various datasets (consisting of tweets of various sizes and non-tweet content, see Table 2) to search for similarities using both *Sequential* search and *SWAPS*. Of interest are the execution times and the number of similar pairs or the mean similarity for both methods. As a form of greedy search, the *Sequential* method always returns a higher proportion of similar items, albeit at the expense of longer execution time. SWAPS, on the other hand, returns a relatively high proportion of similar items more efficiently.

Based on Table 3 and Figure 11, the difference between the items found by the two methods is marginal in SBT and DVT in comparison with other datasets. This is desirable and is probably because the other datasets have been collected and curated for a specific purpose. For example, instances in the *review datasets* exhibit high similarities among them since they are all reviews of products of the same type and, thus, they mostly contain similar terms. Noting the difference in execution time between *Sequential* and *SWAPS*, it could be argued that the difference is not significant enough to warrant compromising accuracy. Considering the window sizes (maximum of 1000 tweets only, see Figure 6), the practical advantage of SWAPS would be appreciated when working with larger window size (as demonstrated in Section 3.3.1).

Figure 11 shows relative proportions of most similar items found by *Sequential* and *SWAPS* methods. The results are the aggregation of the different similarity scores in a window of tweets of variable sizes. Figure 6 shows the different window sizes. The result in Figure 11 only captures samples from the window sizes we consider to be adequate (from 100 up to 500) for illustration. The SWAPS achieves higher counts at a few instances, but the overall count is in favour of the *Sequential* method. Moreover, SWAPS is invoked at a specific point when certain criteria are met (see section 3.3);

Table 3: Execution performance across multiple datasets. SC and MC: execution based on single core and multiple cores, respectively. Only MC is available for datasets with a large pairwise size.

| Dataset | Sequential execution time(s) | | $\mu_{sim}$ | SWAPS execution time(s) | | $\mu_{sim}$ |
|---|---|---|---|---|---|---|
| | SC | MC | | SC | MC | |
| DVT | — | 5640 | 0.26 | — | 2340 | 0.21 |
| SBT | — | 4400 | 0.33 | — | 1800 | 0.28 |
| Review tweets1 | 425 | 252 | 0.63 | 145 | 95 | 0.47 |
| Review tweets2 | — | 1572 | 0.85 | 19975 | 761 | 0.64 |
| Review tweets3 | 5.0 | 5.0 | 0.71 | 12 | 5 | 0.59 |
| Eur-Lex | — | 1325 | 0.60 | — | 348 | 0.46 |
| Social Circles | – | 205 | 0.38 | — | 31 | 0.34 |

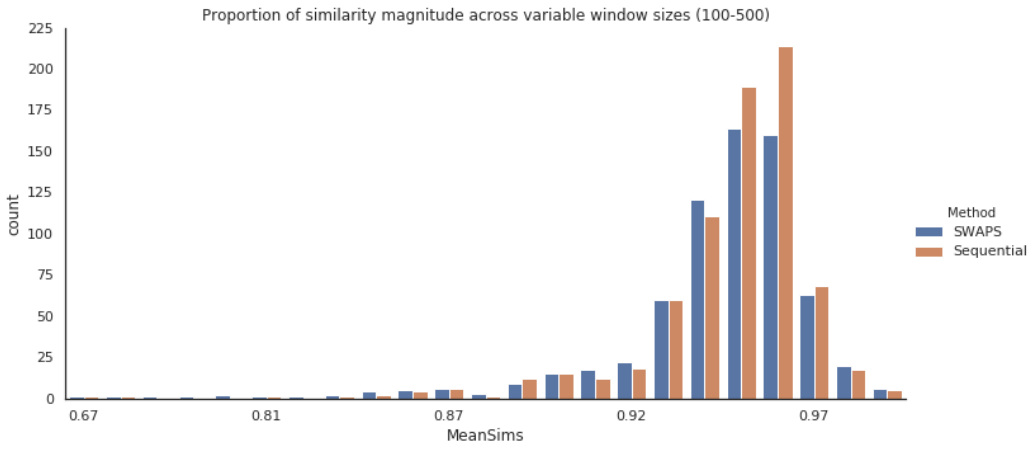


Figure 11: Proportions of mean similarities in searches conducted by *Sequential* and *SWAPS* methods. As expected, the *Sequential* method returns higher proportion of similarity than *SWAPS* but at the expense of longer time (see Table 3). The difference in the proportion between the two is marginal, which can be compromised in favour of speed.

until then, the operation is sequential.

## 4. Tweet's status and choice of anchor tweet

We observe that the *circle size* for anchor tweets differs based on the relevance of the tweet in attracting attention. To demonstrate this, we apply

the concept of light cone[10] and ripples. Figure 12(a) intuitively illustrates the level of interest generated by an anchor tweet. The similarity of a tweet to other tweets increases as the engagement lifespan increases. The similarity of a tweet to other tweets increases as the engagement lifespan increases. Tweets with high engagement tend to have many features in common (e.g. high indegree or followership). These features are considered responsible for higher engagement and as a proxy for online social status. It follows that the category of tweets with high engagement level will show exhibit a certain pattern of similarity by having many features in common. Sustaining a high engagement lifespan, i.e. more ripples in Figure 12(a), is explained by a latent variable we refer to the ability of tweet's relevance which is the relevance of a tweet based on a combination of features defining its status. This phenomenon is investigated by applying the idea of item response theory [9]. With respect to this study, item response theory (IRT) measures the influence of an anchor tweet in attracting more tweets i.e. *is the tweet from a user who has a large following or who tends to have a low favourite counts?* Accordingly, we apply the *Rasch model* to assess the relevance of an anchor tweet, as shown in Figure 12(b).

### *4.1. Engagement level and maximisation*

With a growing data stream and high demand for instant processing where efficiency is crucial, the sequential method is not only time consuming, but computationally expensive. The ultimate goal of the prediction model is to estimate the expected time at which to anticipate a high level of engagement with a tweet. The level of engagement can be discerned since the longer the tweet can attract attention (more *circle members*) the more engaged is the tweet. In Figure 12(a), a more extended period and a large number of circle members are considered high engagement. We sample some tweets with a high number of circles and observe the exact period or time of the day (see Table A.4 for some examples). A substantial number of tweets appear to be produced at a definite period, mostly toward the end of the evening. This behaviour was previously observed and termed *pointless bubbles* [3]. In some cases, most similar tweets (assumed random) are posted around 10am – 4pm and evening period, perhaps due to a large number of

[10]Leveraging the concept of a light cone from Physics and how ripples are created in a pond, in proportion to the surface area.

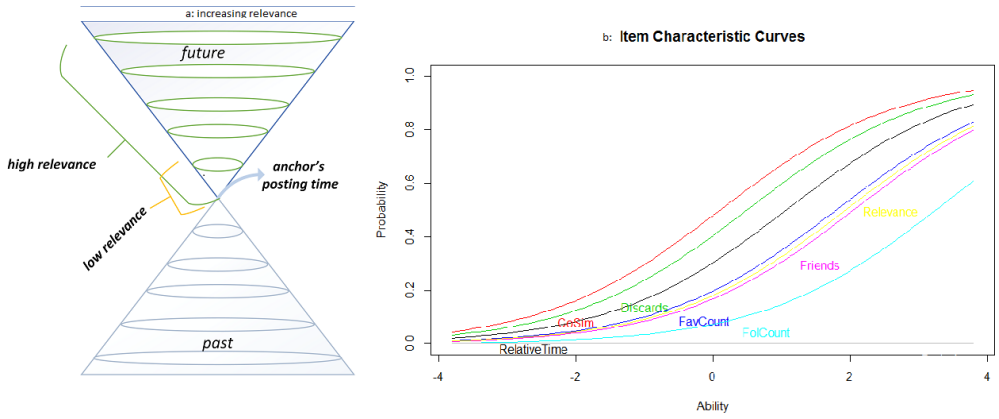


Figure 12: Sub-figure (a) shows how anchor tweet generates attention after posting and observing how the wavelength of each ripple differs. This can be explained by the status of the tweet, e.g. if the tweet was posted by a celebrity or any user with high followership base, it has the potential of attracting interest, thereby generating more ripples before dying out. Sub-figure (b) shows how IRT can measure the ability of a tweet to attract interest. The attributes in the figure constitute the status of a tweet and define its relevance in terms of engagement magnitude or more ripples in sub-figure (a). Higher values denote a more attractive tweet, and the lower categories imply, the higher chances all tweets will possess.

tweets are produced within this period i.e. in comparison with other times of the day. Leveraging this insight and the idea of *tweet's status* can be a useful strategy from a *search* and *ads* viewpoint, among other benefits. Indeed, this is a complex phenomenon that requires many factors (such as collection window, topical discourse and other implicit factors) to be accounted for to improve searches. Other key factors capable of enhancing similarity computation irrespective of the similarity metrics are the *popularity of topic* and the *posting period*.

## 5. Conclusion and Future work

The flexible roles of users as both producers and consumers of content in modern social media are empowering as well as posing many challenges regarding efficient access to relevant information. This paper presents a deep learning strategy based on the idea of a tweet's footprint to improve search and navigation in social media platforms and an efficient searching algorithm. Our approach circumnavigates the challenges in the time-consuming

sequential search for similar items on Twitter by ensuring less search space and improved efficiency. We demonstrate a pragmatic approach to study the distribution and patterns of similar and dissimilar tweets by considering various bootstrap samples drawn from a collection of tweets. We quantify the associated uncertainties and offer useful insights for practical applications. We show how window size affects the distribution of similarity. Increasing the window size to 1000+ was shown to result in high numbers of similar tweets but that 400 - 500 is adequate, especially when the content is about related topics. Concerning SBT and DVT, the window size often spans up to 800 and 1500 respectively. This research is underpinned by statistical evidence which strengthens the validity of the findings. Amongst other benefits, our technique can be applied to various application domains such as topic tracking and detection, clustering and ads.

*Future work.* The proposed SWAPS algorithm balances the trade-off between speed and accuracy, which may omit some relevant items and compromise performance. To maximise the algorithm's functionality, future work will focus on deep reinforcement learning (DRL) to utilise the algorithm in crafting a policy to be utilised by DRL agent. As influential Twitter users promote the exponential growth of particular topics, it becomes challenging to search for less popular topics. The platform then becomes biased towards those influential users. Future research will allow an understanding of the most appropriate time to analyse data from a wide range of specific sets of users, not only the most famous or prolific ones.

## Acknowledgements

The authors would like to thank Prof. Franco Rizzuto for the fruitful discussions and exchange of ideas about a multitude of aspects related to social media, spam content and the motives of spammers. The third author has participated in this research work as part of the *CROSSMINER* Project, which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 732223.

## Reference

## References

[1] Charu C Aggarwal and Karthik Subbian. 2012. Event detection in social streams. In *Proceedings of the 2012 SIAM international conference on*

*data mining*. SIAM, 624–635.

[2] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. 1993. Efficient similarity search in sequence databases. In *International conference on foundations of data organization and algorithms*. Springer, 69–84.

[3] Pear Analytics. 2009. Twitter study. (2009).

[4] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*. 730–738.

[5] Ana Catarina Calheiros, Sérgio Moro, and Paulo Rita. 2017. Sentiment classification of consumer-generated online reviews using topic modeling. *Journal of Hospitality Marketing & Management* 26, 7 (2017), 675–693.

[6] Chun Chen, Feng Li, Beng Chin Ooi, and Sai Wu. 2011. Ti: an efficient indexing mechanism for real-time search on tweets. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 649–660.

[7] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 167–176.

[8] Paul Dourish and Matthew Chalmers. 1994. Running out of space: Models of information navigation. In *Short paper presented at HCI*, Vol. 94. 23–26.

[9] Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.

[10] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S Yu, and Hongjun Lu. 2005. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 181–192.

[11] Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. 2018. Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning. In *Proceedings of the 2018 International Conference on Digital Health*. ACM, 121–125.

[12] Adrien Guille and Cécile Favre. 2015. Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach. *Social Network Analysis and Mining* 5, 1 (2015), 18.

[13] Jonathan Haynes and Igor Perisic. 2009. Mapping search relevance to social networks. In *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*. ACM, 2.

[14] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. 2008. Can social bookmarking improve web search?. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 195–206.

[15] Isa Inuwa-Dutse. 2018. Modelling Formation of Online Temporal Communities. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 867–871.

[16] Isa Inuwa-Dutse, Mark Liptrott, and Ioannis Korkontzelos. 2018. Detection of spam-posting accounts on Twitter. *Neurocomputing* 315 (2018), 496–511.

[17] Amir Karami, Aryya Gangopadhyay, Bin Zhou, and Hadi Kharrazi. 2018. Fuzzy approach topic discovery in health and medical corpora. *International Journal of Fuzzy Systems* 20, 4 (2018), 1334–1345.

[18] Elihu Katz, Paul F Lazarsfeld, and Elmo Roper. 2017. *Personal influence: The part played by people in the flow of mass communications*. Routledge.

[19] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 137–146.

[20] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[21] Cédric Lagnier, Ludovic Denoyer, Eric Gaussier, and Patrick Gallinari. 2013. Predicting information diffusion in social networks using content and user's profiles. In *European conference on information retrieval*. Springer, 74–85.

[22] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. `http://snap.stanford.edu/data`.

[23] Chenliang Li, Aixin Sun, and Anwitaman Datta. 2012. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 155–164.

[24] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 115–124.

[25] Brian Lott. 2012. Survey of keyword extraction techniques. *UNM Education* 50 (2012).

[26] Hsin-Min Lu and Chien-Hua Lee. 2015. A twitter hashtag recommendation model that accommodates for temporal clustering effects. *IEEE Intelligent Systems* 30, 3 (2015), 18–25.

[27] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

[28] Bruce H Mayhew. 1984. Chance and necessity in sociological theory. *Journal of Mathematical Sociology* 9, 4 (1984), 305–339.

[29] Eneldo Loza Mencia and Johannes Fürnkranz. 2008. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 50–65.

[30] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[31] David R Millen and Jonathan Feinberg. 2006. Using social tagging to improve social navigation. In *Workshop on the Social Navigation and Community based Adaptation Technologies*. Citeseer.

[32] Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2, 1–2 (2008), 1–135.

[33] Jie Peng, Craig Macdonald, Ben He, Vassilis Plachouras, and Iadh Ounis. 2007. Incorporating term dependency in the DFR framework. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 843–844.

[34] Steven T Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences* 108, 9 (2011), 3526–3529.

[35] Davood Rafiei and Alberto Mendelzon. 1998. Efficient retrieval of similar time sequences using DFT. *arXiv preprint cs/9809033* (1998).

[36] Gerard Salton. 1971. The SMART retrieval system-experiments in automatic document processing. *Englewood Cliffs* (1971).

[37] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. 2016. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science* 2 (2016), e55.

[38] Narayanan Sundaram, Aizana Turmukhametova, Nadathur Satish, Todd Mostak, Piotr Indyk, Samuel Madden, and Pradeep Dubey. 2013. Streaming similarity search over one billion tweets using parallel locality-sensitive hashing. *Proceedings of the VLDB Endowment* 6, 14 (2013), 1930–1941.

[39] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[40] Rebekah Tromble, Andreas Storz, and Daniela Stockmann. 2017. We Don't Know What We Don't Know: When and How the Use of Twitter's Public APIs Biases Scientific Inference. (2017).

[41] Michail Vlachos, Christopher Meek, Zografoula Vagena, and Dimitrios Gunopulos. 2004. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 131–142.

[42] Duncan J Watts and Peter Sheridan Dodds. 2007. Influentials, networks, and public opinion formation. *Journal of consumer research* 34, 4 (2007), 441–458.

[43] Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 177–186.

[44] Chih-Kuan Yeh, Wei-Chieh Wu, Wei-Jen Ko, and Yu-Chiang Frank Wang. 2017. Learning Deep Latent Space for Multi-Label Classification.. In *AAAI*. 2838–2844.

[45] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).

[46] Arkaitz Zubiaga. 2012. Enhancing navigation on wikipedia with social tage. *arXiv preprint arXiv:1202.5469* (2012).

## Appendix A. Some examples

Table A.4: Example of anchor tweets and corresponding circle members from DVT.

| Type | Text | Posting time |
|---|---|---|
| Anchor | **why did two of the biggest wrestlers in the world pull out of the event in saudi arabia?** | **[2018-11-02 19:56:30]** |
| Circles | foreign office wants to for crimes against journalists. first step - free the journalist detained | [2018-11-02 19:56:30] |
| Circles | the best way to screw the saudi government out of their money? give them and make them like it | [2018-11-02 19:56:34] |
| Circles | this the series of bjp biggies who have used foul language against congress leaders in front of media | [2018-11-02 19:56:33] |
| Circles | perfect eression of the circularity of us policy toward the saudi-iranian cold war | [2018-11-02 19:56:35] |
| Circles | erdogan says that the order to assassinate khashoggi came from the highest level of the saudi regime | [2018-11-02 19:56:35] |
| Anchor | **hulk hogan returns - brock wins the universal title - shawn michaels still has it - + year olds main event** | **[2018-11-02 19:56:57]** |
| Circles | president of turkey recep tayyip rrdogan writes in oped: saudi arabia still has many questions to answer about jamal | [2018-11-02 19:57:00] |
| Circles | president of turkey recep tayyip rrdogan writes in oped: saudi arabia still has many questions to answer about jamal | [2018-11-02 19:57:00] |
| Circles | hulk hogan coming out to real american in saudi arabia is absolutely hysterical | [2018-11-02 19:56:58] |
| Circles | i think the role of the uk govt is not only in bilateral relationships with other governments but also on the ground | [2018-11-02 19:57:01] |
| Anchor | **how many mexican journalists have been slaughtered by the powers that be? yet you say nothing. the same** | **[2018-11-02 19:56:24]** |
| Circles | i'm having a blast watching journalists scramble trying to desperately elain away the robert rourke campaign | [2018-11-02 19:56:25] |
| Circles | i have never felt so much hatred and bigotry in my life. every day the us media is attacking; to solve the shia issue in nigeria, there is a need for frank discussion between the govt imn leadership | [2018-11-02 19:56:26] |
| Circles | i wait for the day that the saudi arabian money of is shown to have bought the blind eyes of fifa | [2018-11-02 19:56:27] |

## Appendix B. Supplementary Figures

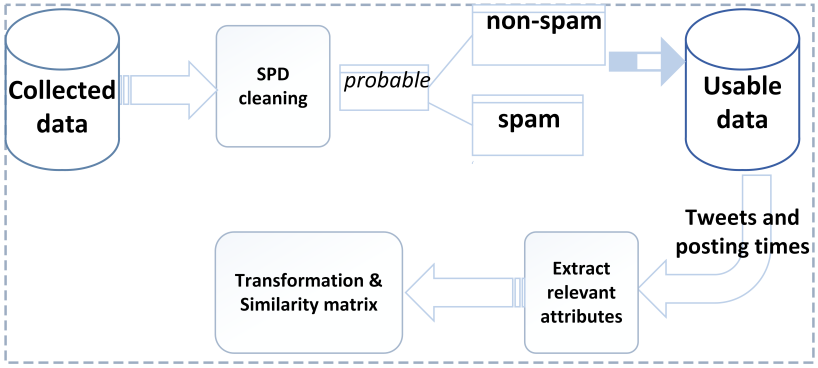


Figure B.13: Summary of the process workflow depicting the data cleaning and transformation phases

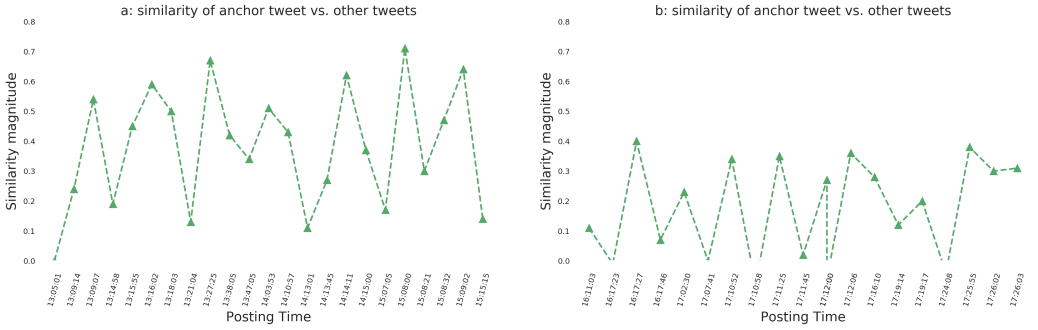


Figure B.14: Splitted Figure 1 in which the peaks denote tweets similar to the anchor tweet where (a) shows period with higher proportion of similar tweets in comparison to (b) with less similar tweets.

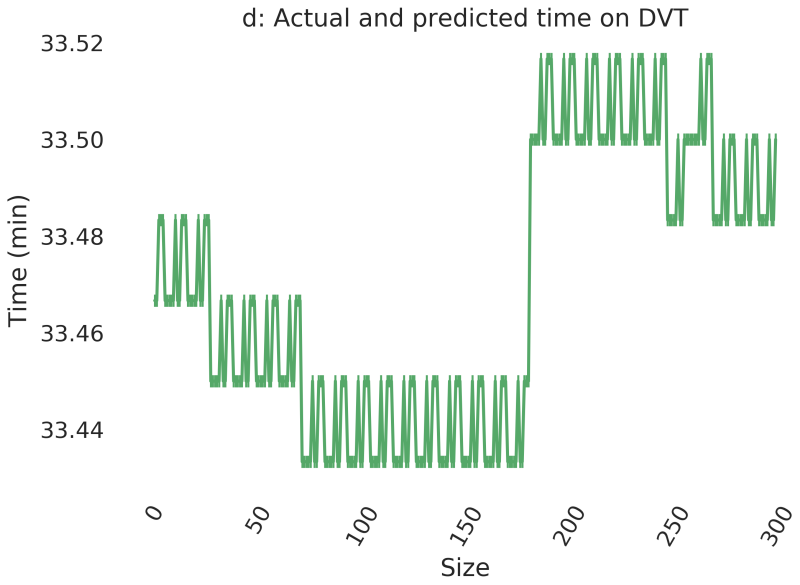


Figure B.15: Expansion of Figure 10(d) to clearly show the period