The Impact of Longstanding Messages in Micro-Blogging Classification

Joana Costa*, Catarina Silva*†, Mário Antunes*‡, Bernardete Ribeiro†
*School of Technology and Management, Polytechnic Institute of Leiria, Portugal
{joana.costa,catarina,mario.antunes}@ipleiria.pt
†Center for Informatics and Systems, University of Coimbra, Portugal
{joanamc,catarina,bribeiro}@dei.uc.pt
‡Center for Research in Advanced Computing Systems, INESC-TEC, University of Porto, Portugal
mantunes@dcc.fc.up.pt

***Abstract*—Social networks are making part of the daily routine of millions of users. Twitter is among Facebook and Instagram one of the most used, and can be seen as a relevant source of information as users share not only daily status, but rapidly propagate news and events that occur worldwide. Considering the dynamic nature of social networks, and their potential in information spread, it is imperative to find learning strategies able to learn in these environments and cope with their dynamic nature.**

Time plays an important role by easily out-dating information, being crucial to understand how informative can past events be to current learning models and for how long it is relevant to store previously seen information, to avoid the computation burden associated with the amount of data produced.

In this paper we study the impact of longstanding messages in micro-blogging classification by using different training time-window sizes in the learning process. Since there are few studies dealing with drift in Twitter and thus little is known about the types of drift that may occur, we simulate different types of drift in an artificial dataset to evaluate and validate our strategy. Results shed light on the relevance of previously seen examples according to different types of drift.

I. Introduction

Twitter is a micro-blogging service where users are able to post text-based messages up to 140 characters, also known as *tweets*. It is also considered one of the most relevant social network, along with *Facebook*, as millions of users are connected to each other by a following mechanism that allows them to read each others posts. The importance of *Twitter* is measured not only by the number of members but, as a consequence of that, by the recognition of entities like governments, brands or news agencies to maintain *Twitter* accounts in order to easily communicate with their fellows.

In its essence *Twitter* is used by individuals to easily share with family and friends their daily activities. However, this concept has evolved and nowadays it is considered a relevant communication platform, as radio or television were, almost in exclusive, a few years ago. Besides the traditional sharing of daily routines, users might share information of broad interest, for instance when multiple users report an event like an earthquake or a terrorist attack. There are a wide range of applications like event detection [1]–[4], academic tool [5]–[7], news media [1], [8] or mining political opinion [9], [10].

Twitter is also responsible for the popularization of the concept of *hashtag*. An *hashtag* is a single word started by the symbol "#" that is used to classify the message content and to improve search capabilities. Although it was popularized in *Twitter*, it is now being widely used, and it has been adopted not only by other social networks like *Facebook* or *Instagram*, but also by other platforms, like television, in order to bridge with their online content. The classification of a *tweet* is particularly important considering the amount of data produced in *Twitter*. Besides that, if we are able to suggest an hashtag for a given *tweet*, we are able to bring a wider audience into discussion [11], spread an idea [12], get affiliated with a community [13], or bring together other Internet resources [14]. Considering the above, it is relevant to study the possibility of identifying an *hashtag* based on the message contents, i.e., if it is feasible to predict the *hashtag* based on the message content.

Hashtag prediction is important, but learning in the *Twitter* environment is not an easy task and requires specific approaches, not only because of the amount of data produced, but also due to its dynamic nature. As *tweets* are organized in a time descending order users tend to perceive as more relevant the newly posted material. When we scale the posted material to millions of users, we realize that time plays an important role, by easily and fast out-dating information. Moreover, in this extremely dynamic environment, concepts appear and reappear, as users concentrate their focus to newly occurring events, forgetting old ones forever or during an unpredictable amount of time. For instance, during a terrorist attack there is a sudden burst of messages related to it during a few days that will probably fade as time passes by, but might reappear if newly information appears, or if it has passed one year or a decade and the event is mentioned again.

To learn in this environment, and considering how infeasible might be using all the data produced, it is essential to understand how informative past events can be to current learning models. This influences for how long it is relevant to store previously seen information, to reduce the computation

burden.

In this paper we study the impact of longstanding messages in micro-blogging classification by using different training time-windows sizes in the learning process. We have also built an artificial dataset by simulating different types of drift as it is unknown the types of drift that occur in *Twitter* real scenario. The obtained results show the relevance of previously seen examples according to different types of drift.

The rest of the paper is organized as follows. We start in Section II by describing the related work regarding social networks and learning in dynamic environments. In Section III we detail our proposed approach, and then proceed in Section IV with the explanation of the experimental setup, including the dataset description, the pre-processing methods, learning and evaluation approaches. In Section V we present and analyse the results obtained. Finally, in Section VI we present the most relevant conclusions and delineate some directions for future work.

II. Related Work

Social networks have gained significant importance and are being widely studied in many fields in the last years. Modern challenges in social networks involve not only computer science issues but also social, political, business, and economical sciences. In computer science, and considering our focus on Twitter, recent works comprise event detection [2], [3], information spreading [15], community mining [16], crowdsourcing [17] and sentiment analysis [10].

In [18] we have proposed the use of meta-classes to boost the performance of Twitter messages classification. This preliminary study shows the possibility of evaluating message content in order to predict *hashtags*. Regarding Twitter *hashtags*, and particularly *hashtag* recommendation, we have also identified the recent study presented in [19], where an approach for *hashtag* recommendation is introduced. This approach computes a similarity measure between *tweets* and uses a ranking system to recommend *hashtags* to new *tweets*. In [20] the use of *hashtags* to classify Twitter messages is done by clustering similar *tweets* in a graph based collective classification strategy. The presented results are promising, despite the fact that this is not an adaptive strategy. A different approach is proposed in [21], where an event detection method is described to cluster Twitter *hashtags* based on semantic similarities between the *hashtags*. This work is in line with our previous work except for the fact that the semantic similarities are computed based on the message content similarities rather than being based on semantic *hashtag* similarities.

Regarding learning in *Twitter*, one must consider the presence of drift. The learning task requires specific approaches, because differently from in commonly used approaches, not all instances contribute equally to the final concept [22]. In non-stationary environments like the Twitter stream, effective learning requires a learning algorithm with the ability to detect context changes without being explicitly informed about them, quickly recover from the context change and adjust its hypothesis to the new context. It should also make use of previous experienced situations when old contexts and corresponding concepts reappear [23]. According to [24], there are four types of drift, namely *sudden*, *gradual*, *incremental* and *reoccurring*. They are represented in Fig. 1.

[25] identified three approaches to handle concept drift: (1) instance selection, (2) instance weighting and (3) ensemble learning. A review of concept drift applied to intrusion detection can be found in [26].

In [27] the algorithm Learn++.NSE is proposed as an algorithm to deal with drift. It learns from consecutive batches of data without making any assumptions on the nature or rate of drift. It learns from environments that experience constant or variable rate of drift, addition or deletion of concept classes, as well as cyclical drift. To deal with scenarios of imbalanced data, the authors in [28] introduce the Learn++.NIE and the Learn++.CDS as two new members of the Learn++ family of incremental learning algorithms that explicitly and simultaneously address the aforementioned phenomena. Learn++.CDS is a combination of the Learn++.NSE algorithm with the SMOTE algorithm proposed by [29]. A different ensemble method called DWM-WIN was recently proposed in [30], to overcome the known limits of [31] namely not considering the time classifiers were define nor the past correct classifications.

We also proposed in [32] three different models to learn in dynamic environments: a time-window model, an ensemble-based model and an incremental model. It is a preliminary study in where we were able to identify whose might be the learning characteristics that are needed to learn in this environment, despite the fact that the time-window model we proposed is unable to retain any past information, the ensemble-based model combines time-window models and the incremental model must store all the information gathered.

Recent important works in the field include [33]–[35]. In [33] authors present an adaptive classifier that exploits both supervised and unsupervised data to monitor the process stationarity. The classifier follows the just-in-time approach and relies on two different change-detection tests to reveal changes in the environment and reconfigure the classifier accordingly. In [34] is presented a theoretically supported framework for active learning from drifting data streams and three active learning strategies are developed for streaming data that explicitly handle concept drift. They are based on uncertainty, dynamic allocation of labeling efforts over time, and randomization of the search space. And finelly, in [35] the authors introduce compacted object sample extraction (COMPOSE), a computational geometry-based framework to learn from nonstationary streaming data, where labels are unavailable (or presented very sporadically) after initialization.

The related work presented so far highlights the importance of dealing with concept drift specially in dynamic scenarios like social networks, and particularly in Twitter, where important information can be mined. Multiple applications like spam email filtering, intrusion detection, recommendation systems, event detection or improve search capabilities are just pointed examples.

III. Proposed Approach

This section describes the proposed approach to study the effect of longstanding messages in micro-blogging classification. We will firstly formalize the problem, a twitter classification problem, and then the proposed strategy we used to define different training time windows.

Twitter classification is a multi-class problem that can be cast as a time series of tweets. It consists of a continuous sequence of instances, in this case, Twitter messages, represented as $\mathcal{X} = \{x_1, \ldots, x_t\}$, where x_1 is the first occurring instance and x_t the latest. Each instance occurs at a time, not necessarily in equally spaced time intervals, and is characterized by a set of features, usually words, $\mathcal{W} = \{w_1, w_2, \ldots, w_{|\mathcal{W}|}\}$. Consequently, instance x_i is denoted as the feature vector $\{w_{i1}, w_{i2}, \ldots, w_{i|W|}\}$.

When x_i is a labelled instance it is represented as the pair (x_i, y_i), being $y_i \in \mathcal{Y} = \{y_1, y_2, \ldots, y_{|\mathcal{Y}|}\}$ the class label for instance x_i.

We have used a classification strategy previously introduced in [18], where the Twitter message *hashtag* is used to label the content of the message, which means that y_i represents the *hashtag* that labels the twitter message x_i. Even though this classification strategy may seem naive, as we are unable to guaranteed that all messages are correctly classified by their *hashtags*, it also seems to be one of the most promising heuristics.

The purpose of this classification problem is to define the unknown predict function $h^t : \mathcal{X} \rightarrow \mathcal{Y}$, that predicts the class label y_i, the *hashtag*, according to x_i, the twitter message. In a time line perspective, h^t uses the historical data $\{x_1, \ldots, x_t\}$ to predict x_{t+1}. The function h^t is then the Twitter message classifier used to predict the *hashtag* of the set of *tweets* presented in the subsequent time windows.

Notwithstanding the twitter message classification is a multi-class problem in its essence, it can be decomposed in multiple binary tasks in a one-against-all binary classification strategy. In this case, a classifier h^t is composed by $|Y|$ binary classifiers.

In order to perceive the importance of past examples in the classification process we present a batch learning model that retains previously seen examples during a defined period. By retaining examples during different periods we aim to evaluate for how long it is relevant to keep information according to the different types of drift, and thus best tailoring the memory mechanism needed for classification purposes.

Algorithm 1 defines the basic steps of our learning model. For each collection of documents $\mathcal{T}$ in a time-window *t*, $\mathcal{T}^t = \{x_1, \ldots, x_{|T^t|}\}$ with labels $\{y_1, \ldots, y_{|T^t|}\} \rightarrow \{-1, 1\}$, and considering the training window size *j*, the dataset $\mathcal{D}^t$ is updated incrementally if the batch temporal moment satisfies the condition *t - j*. By updating the documents collection $\mathcal{D}^t$ based on a training time window we retain the information during a defined amount of time, discarding the examples that occur before that moment.

Considering *Twitter* as a particular case of a time series, one must devise the classification into proper learning models. When a new collection of documents in the subsequent time window occur, we will create a new learning model as proposed above to classify the newly seen examples.

Algorithm 1: Learning Model

Input:
For each collection of documents $\mathcal{T}$ in a time window *t*, $\mathcal{T}^t = \{x_1, \ldots, x_{|T^t|}\}$ with labels $\{y_1, \ldots, y_{|T^t|}\} \rightarrow \{-1, 1\} \quad t = 1, 2, \ldots$

Training window size *j*

```
for t=1,2,... T do
    if t − j then
        D^t ← D^t ∪ T^t
    end
end
Classifier C^t : Learn (D^t), obtain: h^t: X → Y
Classifier C^t : Classify (T^{t+1}), using: h^t: X → Y
```

We also propose to generate an artificial dataset that simulates different times of drift because it is not known the types of drift that occur in the Twitter real scenario. By artificially inducing different types of drift with controlled features, we intend to mainly focus the identification of the learning characteristics best tailored to deal with them, instead of using a real scenario where one can not guarantee not only the presence of drift but also its correct identification.

The drifts we intend to represent are those proposed by [24], namely *sudden*, *gradual*, *incremental* and *reoccurring*. We extend these four types of drift to ten drifts as we also aspire to simulate more drift patterns. For instance positive gradual and negative gradual, and the normality, by using concepts that occur with the same frequency over time.

The main idea of our dataset is to drift the frequency of the Twitter message classification. Since a Twitter labelled dataset is missing so far, we use the *hashtags* enclosed in the message as the message classification, in an approach we have previously introduced in [18].

IV. Experimental Setup

A. Dataset

The dataset we have defined to evaluate and validate our strategy was carried out by defining 10 different *hashtags* that would represent our drifts, based on the assumption that they would denote mutually exclusive concepts, like *#realmadrid* and *#android*. By trying to use mutually exclusive concepts we intent to avoid misleading a classifier, as two different *tweets* could represent the same concept, and that way introducing a new variable to our scenario that could mislead the possible obtained results. In order to achieve a considerable amount of tweets, and consequently diversity, we have chosen trending *hashtags* like *#syrisa* and *#airasia*. Table I shows the chosen *hashtags* and the corresponding drift they represent. This

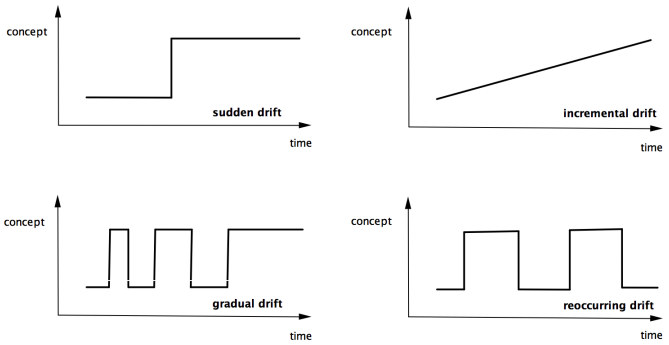

Figure 1. Different types of drift

Drift	Hashtag
Sudden #1	#syrisa
Sudden #2	#airasia
Gradual #1	#isis
Gradual #2	#bieber
Incremental #1	#android
Incremental #2	#ferrari
Reoccurring	#realmadrid
Normal #1	#jobs
Normal #2	#sex
Normal #3	#nfl

Table I
MAPPING BETWEEN TYPE OF DRIFT AND HASHTAG.

correspondence was done arbitrarily and do not correspond to any possible occurrence in the real Twitter scenario, since as stated above, no information is known about the occurrence of drifts in Twitter.

The Twitter API[1] was then used to request public *tweets* that contain the defined *hashtags*. The requests have been cared of between 28 December 2014 and 21 January 2015 and *tweets* were only considered if the user language was defined as English. We have requested more than 75.000 *tweets* concerning the given *hashtags*, even though some of them were discarded, like for instance those *tweets* containing no message content besides the *hashtag*. The *hashtag* was then removed from the message content in order to be exclusively used as the document label. The *tweets* matching this presumptions were considered labelled and suited for classification purposes, and were used by their appearing order in the public feed.

We have simulated the different types of drift by artificially defining timestamps to the previously gathered *tweets*. Time is represented as 100 continuous time windows, in which the frequency of each *hashtag* is altered in order to represent the defined drifts. Each *tweet* is then timestamped so it can belong to one of the time windows we have defined. For instance, Sudden #1 is represented by the appearance of 500 *tweets* with the *hashtag #syrisa* in each time windows from 25 to 32, and in any of the other time windows this *hashtag* appear. Differently from *Sudden #1*, *Sudden #2* is represented with only 200 *tweets* with the *hashtag #airasia* in each time windows from 14 to 31, we tried to simulate a more soft occurring drift, but with a more long-standing appearance. By making both concepts disappear, in time windows, 32 and 31, respectively, we also intended to simulate the opposite way of the [24] proposed sudden drift. Due to space constraints it is unbearable to present a table with the frequency of each *hashtag* in each time window, but it is important to state that *Incremental #2* and *Gradual #2* are represent by the same number of *tweets* in an equal number of time windows, but in a descent way than represented in *Incremental #1* and *Gradual #1* and *Normal #1*, *Normal #2* and *Normal #3* differ in the number of *tweets* that appear in a constant way in all the time windows. Our final dataset contains 34.240 *tweets*.

[1]https://dev.Twitter.com/

B. Representation and Pre-processing

A *tweet* is represented as one of the most commonly used document representation, which is the vector space model, also known as *Bag of Words*. The collection of features is built as the dictionary of unique terms present in the documents collections. Each tweet of the document collection is indexed with the *bag* of the terms occurring in it, i.e., a vector with one element for each term occurring in the whole collection. The weighting scheme used to represent each term is the *term frequency - inverse document frequency*, also know as *tf-idf*.

High dimensional space can cause computational problems in text-classification problems where a vector with one element for each occurring term in the whole connection is used to represent a document. Also, overfitting can easily occur which can prevent the classifier to generalize and thus the prediction ability becomes poor. In order to reduce feature space pre-processing methods were applied. These techniques aim at reducing the size of the document representation and prevent the mislead classification as some words, such as articles, prepositions and conjunctions, called *stopwords*, are non-informative words, and occur more frequently than informative ones. An english-based stopword dictionary was used, but *Twitter* related words like "*rt*" or "*http*" were also considered as they can be seen as *stopwords* in the *Twitter* context. *Stopword removal* was then applied, preventing those non informative words from misleading the classification.

Stemming method was also applied. This method consists in removing case and inflection information of each word, reducing it to the word stem. Stemming does not alter significantly the information included, but it does avoid feature expansion.

	Class Positive	Class Negative
Assigned Positive	a	b
	(True Positives)	(False Positives)
Assigned Negative	c	d
	(False Negatives)	(True Negatives)

Table II
CONTINGENCY TABLE FOR BINARY CLASSIFICATION.

	Training window size									
Drift	1	2	3	4	5	6	7	8	9	10
Sudden #1	92,40%	93,01%	92,99%	92,92%	92,83%	92,80%	92,75%	92,73%	92,69%	92,78%
Sudden #2	90,60%	92,12%	92,70%	93,19%	93,34%	93,36%	93,33%	93,46%	93,43%	93,41%
Gradual #1	52,53%	61,91%	66,22%	68,29%	69,95%	71,00%	73,91%	75,64%	78,49%	80,56%
Gradual #2	74,27%	77,26%	78,94%	78,93%	81,39%	83,82%	85,67%	87,35%	90,10%	91,04%
Incremental #1	83,53%	87,90%	90,37%	91,82%	92,46%	93,11%	93,58%	93,83%	94,08%	94,41%
Incremental #2	60,01%	71,79%	77,17%	80,15%	82,62%	84,41%	85,85%	86,60%	87,52%	88,15%
Reoccurring	54,74%	64,47%	65,17%	64,53%	63,83%	63,08%	62,33%	61,85%	59,14%	58,81%
Normal #1	24,72%	54,03%	66,32%	73,07%	77,03%	79,08%	80,97%	82,78%	83,68%	84,40%
Normal #2	80,07%	87,15%	89,53%	90,87%	91,74%	92,64%	93,40%	93,68%	93,89%	94,11%
Normal #3	42,75%	71,14%	78,03%	82,25%	83,85%	85,40%	86,72%	87,61%	88,12%	88,50%
Average of micro-averaged F_1	71,85%	80,01%	83,16%	84,91%	86,09%	87,04%	87,92%	88,53%	89,10%	89,54%

Table III
MICRO-AVERAGED F_1

C. Learning and Evaluation

The evaluation of our approach was done by the previously described dataset and using the Support Vector Machine (SVM). This machine learning method was introduced by Vapnik [36], based on his Statistical Learning Theory and Structural Risk Minimization Principle. The idea behind the use of SVM for classification consists on finding the optimal separating hyperplane between the positive and negative examples. Once this hyperplane is found, new examples can be classified simply by determining which side of the hyperplane they are on. SVM constitute currently the best of breed kernel-based technique, exhibiting state-of-the-art performance in text classification problems [37]–[39]. SVM were used in our experiments to construct the proposed models.

In order to evaluate a binary decision task we first define a contingency matrix representing the possible outcomes of the classification, as shown in Table II.

In order to evaluate the binary decision task we defined well-known measures based on the possible outcomes of the classification, such as recall ($R = \frac{a}{a+c}$) and precision ($P = \frac{a}{a+b}$), as well as combined measures, such as, the van Rijsbergen F_β measure [40], which combines recall and precision in a single score:

$$F_\beta = \frac{(\beta^2 + 1)P \times R}{\beta^2 P + R}. \quad (1)$$

F_β is one of the best suited measures for text classification used with $\beta = 1$, i.e. F_1, an harmonic average between precision and recall (2), since it evaluates unbalanced scenarios that usually occur in text classification settings and particularly in text classification in the Twitter environment.

$$F_1 = \frac{2 \times P \times R}{P + R}. \quad (2)$$

Considering the proposed approach and the fact that we are working with a time series and we use a one-against all strategy, we will have a classifier for each batch of the time series that is composed by $|Y|$ binary classifiers, being $|Y|$ the collection of possible labels. To perceive the performance of the classification for each drift pattern, we will consider all the binary classifiers that were created in all the time series batches. To evaluate the performance obtained across time, we will average the obtained results. Two conventional methods are widely used, specially in multi-label scenarios, namely macro-averaging and micro-averaging. Macro-averaged performance scores are obtained by computing the scores for each learning model in each batch of the time series and then averaging these scores to obtain the global means. Differently, micro-averaged performance scores are computed by summing all the previously introduces contingency matrix values (*a,b,c and d*), and then use the sum of these values to compute a single micro-averaged performance score that represents the global score.

To evaluate the global performance for each drifting pattern we will use a micro-averaged F_1, that will considered the results obtained for each model created to classify the defined pattern in each batch of the time series. The use of macro-averaged F_1 is discarded because it is impossible to calculate the F_1 measure in all the batches where assigned positives do not exist, as precision is totally dependent on assigned positives. This condition occurs in the batches were a classifier sees a class for the first time, classifying all the newly seen examples as negative examples.

V. EXPERIMENTAL RESULTS

In this Section we evaluate the performance yielded on the Twitter data set using the approach described in Section III. Table III summarises the performance results obtained by classifying the dataset, considering the micro-averaged F_1 measure.

Analysing the table we can observe that in global terms, and considering the average of the micro-averaged F_1, the increase of the training time window size improves the classification. This is normal and expected as the learning models are trained with more examples and this leads to a better performance. Nevertheless, and considering that it is unreasonable to store all the examples for training purposes, it is important to determine the best relation between performance and the computation burden associated with storing and processing the training examples.

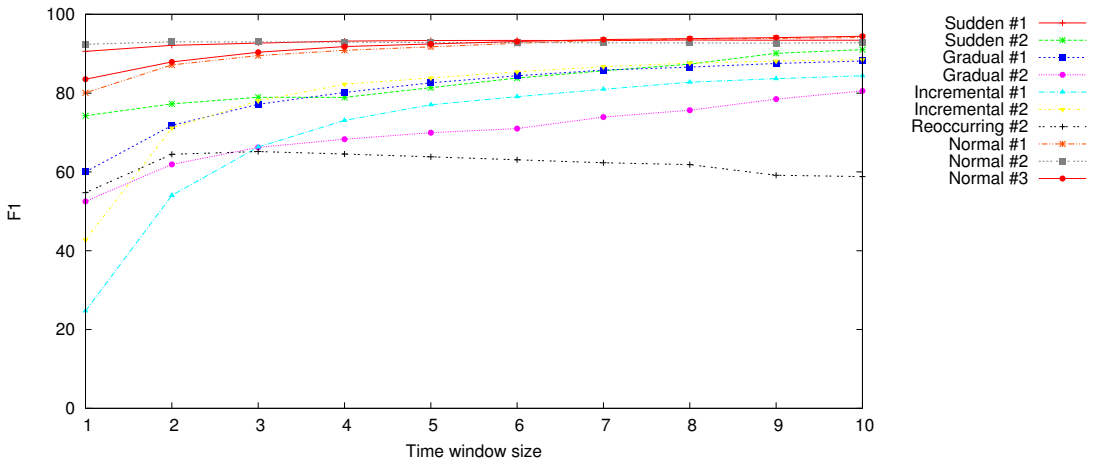

Figure 2. Micro-averaged F_1 obtained by time window

As depicted in Fig. 2, the increase in the average of micro-averaged F_1 seems to slow above training time window size 4, which means that above that value the cost benefit relation is less substantial. It is also important to note the major performance increase from training time window size 1 to training time window size 2. This happens because using two training time windows instead of one implies we that we doubled the training size, which is a major improvement. From then on the proportion is less substantial, for instance from training time window 2 to training time window 3 there is only a 50% increase, while from training time window size 3 to training time window size 4 we have an increase of 33% and so forth. One can argue that this is highly dependent on the computational complexity of processing one time window and no values are shown in this study about this complexity, but we have used similar sized time windows, which means that in proportional terms one can define the time window based on the computational complexity that can handle.

Although the overall performance seems to be increasing with the increase of the training time window size, there are particular cases in which the performance decreases. Firstly, there are small decreases that are so small, and do not define a decreasing pattern, that one can consider as less significant, like for instance in the drift *Gradual #1* from time window size 3 to time window size 4, respectively 78,94% to 78,93%. Finally there are performance decreases that, small or not, seem to define a pattern that might be related with the nature of the drift pattern that is represented, like *Sudden #1*, *Sudden#2* and *Reoccurring*.

The performance in the identification of *tweets* from *Sudden#1* increases from training time window size 1 to training time window size 2. As explained above this might be related to the doubled size time window in the training phase, but then on, even in small amounts, the performance starts to decrease. As mentioned in the Section IV-A, drift *Sudden #1* occur only in time windows from number 25 to 32, with a constant amount of *tweets* per time window, 500 in each time window. As an example, to classify the examples in time window 27, with training time window size 2, the classifier receives as training examples all the examples from time windows 25 and 26, which means that we sees more positive examples than we would if trained with only examples from time window 26 (in case the time size window is 1). But when we increase the time window to 4, that means that in the training phase the classifiers sees examples from time windows 23, 24, 25 and 26, and in time windows 23 and 24 there are only negative examples, since the drift only starts to happen in the time window 25. This explains why in sudden drifts the increase of the time size window leads to a decrease in the performance, because past events hardly contribute with positive examples as the drift appeared in a sudden way in a specific temporal moment.

It is also noteworthy that in *Sudden #2* the decrease pattern is different. Whilst it is a drift with the same nature, one should expect the same pattern in the classification performance, but the major difference from drift *Sudden #1* to *Sudden #2* is that *Sudden #1* is more abrupt than *Sudden #2*. As referred in Section IV-A, the number of examples that appear in each time window where *Sudden #2* is represented is more than a half that *Sudden #1*, from 500 to 200 *tweets*, but it also happens that *Sudden #2* is much longer in time that *Sudden #1*. As being a much longer drift past events contribute differently to the performance of the classifier, because in *Sudden #2* past windows might more easily contribute with more positive examples than in *Sudden #1*. This attest the importance of the nature of the drift pattern, along with its particular characteristics, in the performance of the classification.

There is also a decrease performance pattern in the drift *Reoccurring*. Firstly, it is importance to explain the characteristics of *Reoccurring* in order to understand the obtained results. *Reoccurring* occur in 5 consequent time windows, for instance from time window 12 to 16 or from time window 28 to 32 and then disappear during 11 time windows. As a consequence, from time window 17 to 27 (both inclusively) there are not positive examples from this class, and also from time window 33 to 43, but in 44 until 48 the drift pattern in

again represented with positive examples. The decrease pattern might be explained because *Reoccurring* can be seen as having the same characteristics of the sudden drifts, specially because it always disappears for 11 time windows. As we do not use in our experiments time window sizes bigger then 10, we do not reach the point in which we provide the classifier with positive examples from the previous burst in which the drift occur, and thus increasing the time window and not reaching that moment could always lead to the same thing that happen with the sudden drift and explained above.

VI. Conclusions and Future Work

In this paper we have studied the impact of longstanding messages in micro-blogging classification. The proposed approach uses different training time windows sizes to understand the possibility of achieving the best balance between the classifier performance and the computational effort needed in the training phase. Since it is not known which types of drift occur in the context of social networks, and particularly in *Twitter*, we have also simulated different types of drift in an artificial dataset to evaluate and validate our strategy.

The results revealed the usefulness of our strategy, specially because it is easy to identify a major slowdown in the increase of performance from training time window size 4 to the subsequent training time window sizes. More precisely, we have identified that there is a major improve from time window size 1 to time window size 2. Even thought in average the increase in the training window size is echoed in an increase in the classification performance, the cost benefit decreases from then on, and specially above time window size 4.

It is also important to conclude the highlights identified concerning the effect of the increase of the training time window size in the classification performance considering drifts with the same nature, specially in cases in which the characteristics of the drift are equal or similar to sudden drift. In these cases, a different strategy must be put forward, as the increase of the training time window size will not always lead to an increase of the classification performance rather than to a decrease, more significant or not depending of the abruptability and the long-standing of the represented drift pattern.

Our future work will include a pruning strategy based on the identification of which are the relevant examples for future classification purposes, and thus reduce the size of the training set by discarding the non relevant ones. We also aim to validate the scalability of our strategy by using distributed computing in a real Twitter scenario.

Acknowledgments

This work is supported by CISUC, via national funding by the FCT - Fundação para a Ciência e a Tecnologia. The iCIS project (CENTRO-07-ST24-FEDER-002003) is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER.
This work is financed by the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project UID/EEA/50014/2013.

References

[1] F. Abel, Q. Gao, G.-J. Houben, and K. Tao, "Semantic enrichment of twitter posts for user profile construction on the social web," in *Proc. 8th Extended Semantic Web Conference on The Semanic Web*, 2011, pp. 375–389.

[2] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proc. 19th Int. Conference on World Wide Web*, 2010, pp. 851–860.

[3] A.-M. Popescu and M. Pennacchiotti, "Detecting controversial events from twitter," in *Proc. 19th Int. Conference on Information and knowledge management*, 2010, pp. 1873–1876.

[4] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: real-world event identification on twitter," in *Proc. 5th Int. Conference on Weblogs and Social Media*, 2011.

[5] S. Ovadia, "Exploring the potential of Twitter as a research tool," *Behavioral & Social Sciences Librarian*, vol. 28, pp. 202–205, 2009.

[6] M. Stankovic, M. Rowe, and P. Laublet, "Mapping tweets to conference talks: A goldmine for semantics," in *Proc. 3rd Int. Workshop on Social Data on the Web*, 2010.

[7] K. Weller and C. Puschmann, "Twitter for scientific communication: how can citations/references be identified and measured?" in *Proc. of the Web Science Conference*, 2011, pp. 1–4.

[8] F. Abel, Q. Gao, G.-J. Houben, and K. Tao, "Analyzing user modeling on twitter for personalized news recommendations," in *Proc. 19th Int. Conference on User Modeling, Adaption, and Personalization*, 2011, pp. 1–12.

[9] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proc. 4th Conference on Weblogs and Social Media*, 2010, pp. 178–185.

[10] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: linking text sentiment to public opinion time series," in *Proc. Int. Conference on Weblogs and Social Media*, 2010.

[11] S. Johnson, "How twitter will change the way we live," *Time Magazine*, vol. 173, pp. 23–32, 2009.

[12] O. Tsur and A. Rappoport, "What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities," in *Proc. 5th Int. Conference on Web Search and Data Mining*, 2012, pp. 643–652.

[13] L. Yang, T. Sun, M. Zhang, and Q. Mei, "We know what @you #tag: does the dual role affect hashtag adoption?" in *Proc. 21st Int. Conference on World Wide Web*, 2012, pp. 261–270.

[14] H.-C. Chang, "A new perspective on twitter hashtag use: diffusion of innovation theory," in *Proc. 73rd Annual Meeting on Navigating Streams in an Information Ecosystem*, 2010, pp. 85:1–85:4.

[15] B. Doerr, M. Fouz, and T. Friedrich, "Why rumors spread so quickly in social networks," *Communications of ACM*, vol. 55, no. 6, pp. 70–75, 2012.

[16] C. Tantipathananandh and T. Y. Berger-Wolf, "Finding communities in dynamic social networks," in *Proc. 11th Int. Conference on Data Mining*, 2011, pp. 1236–1241.

[17] M. Treiber, D. Schall, S. Dustdar, and C. Scherling, "Tweetflows: flexible workflows with twitter," in *Proc. 3rd Int. Workshop on Principles of Engineering Service-Oriented Systems*, 2011, pp. 1–7.

[18] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "Defining semantic meta-hashtags for twitter classification," in *Proc.11th Int. Conference on Adaptive and Natural Computing Algorithms*, 2013, pp. 226–235.

[19] E. Zangerle, W. Gassler, and G. Specht, "Recommending #-tags in twitter," in *Proc. 19th Int. Conference on User Modeling, Adaptation and Personalization*, 2011, pp. 67–78.

[20] Y. Duan, F. Wei, M. Zhou, and H.-Y. Shum, "Graph-based collective classification for tweets," in *Proc. 21st Int. Conference on Information and Knowledge Management*, 2012, pp. 2323–2326.

[21] H. O. O Ozdikis, P Senkul, "Semantic expansion of hashtags for enhanced event detection in Twitter," in *1st Int. Workshop on Online Social Systems*, 2012.

[22] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Dynamic integration of classifiers for handling concept drift," *Information Fusion*, vol. 9, no. 1, pp. 56–68, 2008.

[23] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996.

[24] I. Zliobaite, "Learning under concept drift: an overview," Vilnius University, Faculty of Mathematics and Informatic, Tech. Rep., 2010.

[25] A. Tsymbal, "The problem of concept drift: definitions and related work," Department of Computer Science, Trinity College Dublin, Tech. Rep., 2004.

[26] J. Kim, P. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, "Immune system approaches to intrusion detection - a review," *Natural Computing*, vol. 6, no. 4, pp. 413–466, 2007.

[27] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Transactions on Neural Networks*, pp. 1517–1531, 2011.

[28] G. Ditzler and R. Polikar, "Incremental learning of concept drift from streaming imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2283–2301, 2013.

[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[30] D. Mejri, R. Khanchel, and M. Limam, "An ensemble method for concept drift in nonstationary environment," *Journal of Statistical Computation and Simulation*, vol. 83, no. 6, pp. 1115–1128, 2013.

[31] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: a new ensemble method for tracking concept drift," in *Proc. 3rd Int. Conference on Data Mining*, 2003, pp. 123–130.

[32] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "Concept drift awareness in twitter streams," in *Proc. 13th Int. Conference on Machine Learning and Applications*, 2014, pp. 294–299.

[33] C. Alippi, G. Boracchi, and M. Roveri, "Just-in-time classifiers for recurrent concepts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 620–634, 2013.

[34] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 27–39, 2014.

[35] K. Dyer, R. Capo, and R. Polikar, "Compose: a semisupervised learning framework for initially labeled nonstationary streaming data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 12–26, 2014.

[36] V. Vapnik, *The Nature of Statistical Learning Theory*, 1999.

[37] T. Joachims, *Learning Text Classifiers with Support Vector Machines*, 2002.

[38] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[39] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "On using crowdsourcing and active learning to improve classification performance," in *Proc. 11th Int. Conference on Intelligent Systems Design and Applications*, 2011, pp. 469–474.

[40] C. van Rijsbergen, *Information Retrieval*, 1979.