A Quantitative Analysis of the Temporal Effects on Automatic Text Classification

Thiago Salles
Department of Computer Sience, Universidade Federal de Minas Gerais, Brazil. E-mail: tsalles@dcc.ufmg.br

Leonardo Rocha
Universidade Federal de São João del-Rei, Brazil. E-mail: lcrocha@ufsj.edu.br

Marcos André Gonçalves
Department of Computer Sience, Universidade Federal de Minas Gerais, Brazil. E-mail: mgoncalv@dcc.ufmg.br

Jussara M. Almeida
Department of Computer Sience, Universidade Federal de Minas Gerais, Brazil. E-mail: jussara@dcc.ufmg.br

Fernando Mourão
Department of Computer Sience, Universidade Federal de Minas Gerais, Brazil. E-mail: fhmourao@dcc.ufmg.br

Wagner Meira Jr.
Department of Computer Sience, Universidade Federal de Minas Gerais, Brazil. E-mail: meira@dcc.ufmg.br

Felipe Viegas
Department of Computer Sience, Universidade Federal de Minas Gerais, Brazil. E-mail: frviegas@dcc.ufmg.br

Automatic text classification (TC) continues to be a relevant research topic and several TC algorithms have been proposed. However, the majority of TC algorithms assume that the underlying data distribution does not change over time. In this work, we are concerned with the challenges imposed by the temporal dynamics observed in textual data sets. We provide evidence of the existence of temporal effects in three textual data sets, reflected by variations observed over time in the class distribution, in the pairwise class similarities, and in the relationships between terms and classes. We then quantify, using a series of full factorial design experiments, the impact of these effects on four well-known TC algorithms. We show that these temporal effects affect each analyzed data set differently and that they restrict the performance of each considered TC algorithm to different extents. The reported quantitative analyses, which are the original contributions of this article, provide valuable new insights to better understand the behavior of TC algorithms when faced with nonstatic (temporal) data distributions and highlight important requirements for the proposal of more accurate classification models.

Introduction

Text classification is one of the major information retrieval problems, and developing robust and accurate classification models continues to be needed as a consequence of the increasing complexity and scale of current application scenarios, such as the web. The task of automatic text classification (TC) aims at creating models that associate documents with semantically meaningful categories. These models are key components for supporting and enhancing a variety of other tasks such as automated topic tagging (i.e., assigning labels to documents), building topic directories, identifying the writing style of a document, organizing digital libraries, improving the precision of web searching, and even helping users to interact with search engines.

TC is a supervised machine learning technique where a set of already classified documents (training set) is used to

Received March 16, 2014; revised September 16, 2014; accepted October 13, 2014

• Published online 7 August 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.23452

build a classifier. The classifier is then employed to predict the classes of unclassified (test) documents. A fundamental TC task is to determine a set of characteristics that better identify a (previously unclassified) document's class. The majority of supervised algorithms consider that all (preclassified) documents provide equally important information to determine such characteristics. However, such a premise may not hold in practice due to several factors, such as its creation time, the venue in which a document was published, and its authors, among other factors.

In this work, we are particularly concerned with the impact that *temporal effects* may have on TC algorithms. That is, the characteristics of a textual data set may change as time goes by, due to several factors, such as the dynamics of knowledge and even language. For example, the percentage of documents belonging to different classes may change as a consequence of the so-called virtual concept drift (Tsymbal, 2004). Thus, density-based classifiers, which are sensitive to class distribution, may not work well because the class distribution from the training set may not be the actual class distribution when the test document was created (Yang & Zhou, 2008; Zhang & Zhou, 2010). Both the temporal variations in class frequencies and the relationships between terms and classes may affect classification effectiveness. That is, the distribution of terms among classes may vary over time due to changes in writing style, term usage, and so on. Therefore, the temporal dynamics of the data are an important aspect that should be taken into account while building more accurate classification models.

Specifically, we consider a *batch classification setting*, in which a fixed set of training documents is available to build a classifier. Such a training set, however, may span over several moments. Let $\mathbb{C}$ and $\mathbb{P}$ denote the set of classes and creation times, respectively, observed in the training set. We represent each training document x as a triple $\langle \vec{x}, c, p \rangle$, where $\vec{x}$ is the vector (bag of words[1]) representation of x, $c \in \mathbb{C}$ denotes its class, and $p \in \mathbb{P}$ its creation time. Furthermore, unlike typical concept drifting formulations, we do not assume any particular temporal ordering among documents (both when training and classifying new documents). Several real-world applications do not impose any temporal ordering over documents. Examples of such applications include the automatic classification of crawled web-pages (which may be older than previously crawled, and labeled, web-pages), and the automatic classification of documents belonging to digital libraries (when incorporating "old" documents to them).

Under this batch setting, Mourão et al. (2008) identified three different temporal effects that may affect the performance of automatic classifiers. The first effect is the *class distribution variation*, which accounts for the impact of the temporal evolution on the relative frequencies of the classes. The second effect is the *term distribution variation*, which refers to changes in the terms' representativeness with respect to the classes as time goes by. The third effect is the *class similarity variation*, which accounts for how the similarity among classes, as a function of the terms that occur in their documents, changes over time. The authors showed that accounting for the temporal evolution of documents poses challenges to building classification models. The learning is usually less effective when such factors are neglected, as assumptions made when the model is built (i.e., learned) may no longer hold due to temporal effects. In fact, traditional TC algorithms, such as K Nearest Neighbors (KNN) Yang (1994) and Support Vector Machine (SVM) Joachims (1999), neglect the temporal dimension, as they make use of just the content and class information when learning the classification model. We argue that one may improve the classification effectiveness by properly using the temporal information to circumvent the negative impact of temporal effects (as shown in Salles et al., 2010).

Despite these previous studies, to the best of our knowledge a thorough analysis of how and to what extent these temporal effects really impact TC algorithms has not been performed yet. A key aspect to be addressed in this task concerns the peculiar behavior that each temporal effect may present in different data sets. For example, although some data sets may present large class distribution variations over time, other data sets may, in contrast, present a more significant variability on term distribution. Moreover, different TC algorithms may be distinctly affected by these effects due to their sensitivity or robustness to each specific effect. In other words, the best strategy to handle temporal effects may depend on the specific characteristics of both, the data set and the used TC algorithm. This makes the task of learning a more accurate classification model that deals with these effects, a challenging one. The purpose of this work is to shed more light on the existence of such temporal effects and to provide a better understanding on how they affect traditional TC algorithms. We here consider traditional TC algorithms, notably Rocchio, Manning, Raghavan, and Schtze (2008), Naïve Bayes, Nigam and McCallum (1998), KNN and SVM, which assume a static data distribution over the training set. We do so for two reasons. First, these traditional classifiers are widely used in batch text classification tasks (Joachims, 1998; Yang and Liu, 1999). Second, because they do not take into account the temporal information, we are able to study their behavior in face of the aforementioned temporal effects. This ultimately gives us a deeper comprehension on how to design better strategies to derive algorithms that are robust to such effects, as well as to understand the strengths and weaknesses of these algorithms when subject to such effects.

In summary, we provide answers to the following two questions: (a) Which are the strongest temporal effects for a given data set? and (b) What is the behavior of each TC algorithm when dealing with different degrees of each temporal effect? Answers to these questions are key to provide a better understanding of the impact of temporal effects on TC. In fact, it has been already established that these

[1] Bag-of-words is the most widely used document representation in TC, and that is why we use it. However, we intend to use more complex representations (see, for instance, Figueiredo et al. [2011] in future studies).

temporal effects do exist in some collections and affect negatively one specific algorithm, namely, the SVM classifier (Mourão et al., 2008). In this work, we take several steps further towards answering the posed questions, by proposing a factorial experimental design (Jain, 1991) aimed at quantifying the impact of the temporal effects on four representative TC algorithms, considering three textual data sets with different characteristics in their temporal evolution.

Thus, the original contributions of this article include: (a) a revisitation of the characterization reported in (Mourão et al., 2008), with the inclusion of a third data set belonging to a distinct and more dynamic domain, in order to strengthen the argument for the existence of such temporal effects; (b) the proposal of a method to enable a deeper and *quantitative* study of the aforementioned temporal effects, by means of a factorial experimental design aimed at uncovering how each temporal effect affects each TC algorithm and textual data set; and (c) an instantiation of that method using several data sets and TC algorithms, along with a detailed study regarding the impact of the temporal effects on them. Specifically, we focus on four traditional TC algorithms, namely, Rocchio, K Nearest Neighbors (KNN), Naïve Bayes and Support Vector Machine (SVM), and on three different and widely used textual collections covering long time periods, namely, ACM-DL (22 consecutive years), MEDLINE (15 consecutive years), and finally, AG-NEWS (573 consecutive days). We should stress that our findings regarding the behavior of such algorithms in the face of these temporal effects have not been reported before and are an original contribution of this work. In fact, our proposed quantitative analysis allows us to rank the temporal effects according to different algorithms and data sets, enabling us to state the most significant ones and their impacts in each context, which is also an original contribution.

As we shall see, temporal effects are stronger on the ACM-DL and AG-NEWS data sets when compared to the MEDLINE data set. In the ACM-DL data set, variations of class distribution and class similarity are statistically equivalent to the variation in term distribution, whereas the first two effects are stronger in MEDLINE and AG-NEWS. These findings motivate the development of strategies to handle the temporal effects in TC algorithms according to each data set's characteristics. Furthermore, the effectiveness of all four analyzed TC algorithms are affected by the aforementioned temporal effects. Indeed, the most significant performance losses were observed when these algorithms were applied to the most dynamic dynamic data sets, ACM-DL and AG-NEWS. Our new analyses also revealed that the SVM classifier is more resilient to the term distribution effect, although also impacted by the other two effects. We also show that the other three algorithms, on the other hand, are very sensitive to all three effects. These results corroborate our argument that the temporal dimension should be considered for the sake of learning accurate classification models.

The next section briefly reviews relevant related work that aims at identifying and/or exploring temporal aspects to improve TC effectiveness. Following that, the next section describes the workload used in our experimental design, that is, the reference data sets and the analyzed TC algorithms. An extension of the characterization performed by Mourão et al. (2008), providing evidence of the existence of temporal effects in three textual data sets, is presented in the next section. Then, the factorial experimental approach proposed is described as a method to provide a more precise picture of the impact of temporal effects on different TC algorithms, whereas the results of applying the proposed methodology on the considered data sets and TC algorithms are discussed in the next section. Finally, the last section concludes the paper and offers some possible directions for future work.

Literature Review

Although TC is a widely studied subject, the analysis of temporal aspects in this class of algorithms is quite recent—it has been studied only in the last decade or so. Most previous studies have focused on detecting and dealing with these effects to improve classification quality, whereas we are aware of only two prior efforts towards characterizing (but not quantitatively) the impact of temporal effects on TC effectiveness. We review previous efforts toward these two directions next.

Detecting and Dealing With Temporal Effects

We start by reviewing previous attempts to detect significant changes in the underlying data distribution due to temporal effects. Gama, Medas, Castillo, and Rodrigues (2004) presented a method to detect changes in the distribution of the training examples by means of an online classifier that performs a sequence of trials to perform the classification. On each trial, it makes some predictions and receives feedback accounting for the classification error in order to detect significant changes in the data at hand. This approach is able to detect both gradual and abrupt changes. Similarly, Nishida and Yamauchi (2009) proposed a system to detect and predict changing distributions by managing a set of offline and online classifiers to account for, respectively, data variations and classifiers' prediction errors. Furthermore, the system also performs a clustering step to allow the prediction of future variations. Other studies explored the use of statistical tests to detect drift (Dries & Rückert, 2009; Nishida & Yamauchi, 2007). In Dries and Rückert (2009), for instance, the authors proposed three adaptive tests that are capable of adapt to different (gradual or abrupt) changing behaviors. In Nishida and Yamauchi (2007), the authors proposed to classify a set of examples belonging to a recent time window, and compared the achieved accuracy against the one obtained with a global classifier that considers all available data. The basic idea is that statistically significant decreases in accuracy suggest data variations. This solution is able to quickly detect drifts when the window size is small, at the cost of being susceptible to data sparseness (because small windows lead to small training sets).

Previous efforts to deal with varying data distributions may be categorized into two broad areas, namely, adaptive document classification and concept drift. Adaptive document classification (Cohen & Singer, 1999) embodies a set of techniques to deal with changes in the underlying data distribution so as to improve the effectiveness of document classifiers through incremental and efficient adaptation of the classification models. Adaptive document classification brings three main challenges to document classification (Liu & Lu, 2002). The first one is the definition of a context and how it may be exploited to devise more accurate classification models. A context is a semantically significant set of documents. Previous research suggests that contexts may be determined through at least two strategies: identification of neighbor terms to a certain keyword (Lawrence & Giles, 1998), and identification of terms that indicate the scope and semantics of the document (Caldwell, Clarkson, Rodgers, & Huxor, 2000). The second challenge is how to build the classification models incrementally (Kim, Park, Deards, & Kang, 2004), whereas the third challenge is associated with the computational efficiency of the resulting classifiers.

The concept or topic drift (Tsymbal, 2004) comprises another relevant set of efforts to deal with varying data distributions in classification. A prevailing approach to address concept drift is to completely retrain the classifier according to a sliding window, which ultimately involves sample selection techniques. A number of previous studies fall into this category. For instance, the method presented in Klinkenberg and Joachims (2000) employs a window with examples sufficiently "close" to the current target concept, and automatically adjusts the window size so that the estimated generalization error is minimized. In Žliobaite (2009), a classification model was built using training examples that are close to the test in terms of both time and space. The methods presented in Klinkenberg (2004) either maintain an adaptive time window on the training data, or select representative training examples, or weight them. Widmer and Kubat (1996) described a set of algorithms that react to concept drift in a flexible way and are able to exploit and can take advantage of the reappearance of contexts. The main idea of these algorithms is to keep only a window of currently trusted samples and hypotheses, and store associated concept descriptions to reuse them if a context reappears. In Rocha, Mourão, Pereira, Gonçalves, and Meira (2008), the authors introduced the concept of temporal context, defined as a portion of the training data set that minimizes the impact of temporal effects on the performance of classifiers. They also proposed an algorithm, named *Chronos*, to identify these contexts based on the stability of the terms in the training set. Temporal contexts are used to sample the training examples for the classification process, and examples considered to be outside the temporal context are discarded by the classifier.

Unlike previous efforts that used a single window to determine drift in the data, Lazarescu, Venkatesh, and Bui (2004) presented a method that uses three windows of different sizes to estimate changes in the data. Although algorithms that use a fixed-size window impose hard constraints on drift patterns, those that use heuristics to adjust the window size to the current duration of the concept drift often require the calibration of several parameters. To provide some theoretical basis for the choice of the window size, Kuncheva and Žliobaite (2009) developed a framework for modeling the classification error as a function of the window size, aiming at determining an optimal window size choice. Such optimal choice leads to statistically significant improvements in window-based strategies. Similarly, Bifet and Gavaldà (2006) proposed a window-based strategy for dealing with drifting data streams that automatically chooses the optimal window size, called ADWIN. This approach keeps a window W with the most recent data, and splits it into two adjacent subwindows W_0 and W_1. Using statistical tests to compare both windows, it detects when a drift occurred. In this case, all possible adjacent subwindows must be considered. Clearly, this is a costly operation (both in terms of time and memory). In Bifet and Gavaldà (2007), the authors proposed an improvement for ADWIN, called ADWIN2, with the same effectiveness guarantees of ADWIN and more efficient data structures.

A second approach to deal with concept drift consists of properly weighting training examples while building the classification model to reflect the temporal variations in the underlying data distribution. The logic for weighting training examples according to some time-based utility function is that window-based approaches are too rigid and may miss valuable information laying outside the window. Following this direction, Koychev (2000) defined a linear time-based utility function to account for variations in the data distribution such that the impact of the examples on the classification model decreases with time. Experimental evaluation employing the algorithms Naïve Bayes and ID3 demonstrated the effectiveness of such approach. In Klinkenberg and Rüping (2003), the authors defined an exponential time-based function to weight examples based on their age. The reported experimental evaluation showed that weighting examples in drifting scenarios lead to significant improvements over fixed-window strategies while being outperformed by an adaptive-window approach. In Kong and Yu (2011), the authors proposed an ensemble of random trees capable of handling stream data with concept drift. Such a result was achieved by means of an exponential weighting function applied to each tree node to gradually reduce the influence of past data in the final classification decision. However, such time-based utility functions are typically defined in a very ad-hoc manner (e.g., linear functions, exponential functions), without any theoretical justification based on changes in data patterns.

Thus, the following question remains unanswered: How can we properly define such a time-based utility function? To answer that question not only should the temporal distance between training and test examples be considered but also the varying characteristics of the underlying data distribution. Salles et al. (2010) presented a statistical analysis of the

DOI: 10.1002/asi

temporal effects on two textual data sets to define a *temporal weighting function* that properly models the changing behavior of the underlying data distribution, reflecting its dynamic nature and capturing both the temporal distance between training and test examples and the variations of the characteristics of the data set. Two instances of weighting strategies that employ the temporal weighting function to deal with these variations were developed and applied to three well-known TC algorithms, namely, Rocchio, KNN, and Naïve Bayes. The experimental results indicated that the temporally aware classifiers achieve statistically significant gains over their traditional counterparts.

Another common approach to deal with concept drift focuses on the combination of various classification models generated from different algorithms (ensembles) for classification, pruning, or adapting the weights according to recent data (Folino, Pizzuti, & Spezzano, 2007; Kolter & Maloof, 2003; Scholz & Klinkenberg, 2007). Scholz and Klinkenberg (2007) proposed a boosting-like method to train a classifier ensemble from data streams. It naturally adapts to concept drift and allows one to quantify the drift in terms of its base learners. The algorithm was shown to outperform learning algorithms that ignore concept drift. In the same direction, Kolter and Maloof (2003) presented a technique that maintains an ensemble of base learners, predicts instance classes using a weighted-majority vote of these "experts," and dynamically creates and deletes experts in response to performance changes. Additionally, Folino et al. (2007) proposed to build an ensemble of classifiers using genetic programming to inductively generate decision trees. In spite of these prior proposals, one important challenge for ensemble classifiers is the efficient management of multiple models.

In this work we do not attempt to provide strategies to circumvent the temporal effects. Instead, we characterize how the temporal effects do indeed manifest on textual data sets and impact automatic text classifiers. Such knowledge is paramount to the development and understanding of effective strategies that try to minimize the impact of these effects on TC.

Characterizing the Temporal Effects

In addition to the aforementioned studies, which aim at either detecting or exploiting the changes in data distribution, in Forman (2006) the author provided a characterization of varying data distributions in the textual data domain, where the concept drift problem is studied considering three main types of data variations: (a) shifting class distribution, which is reflected by the observed variations over time in the proportion of documents assigned to each class; (b) shifting subclass distribution, which accounts for varying feature distributions; and finally (c) the fickle concept drift, which denotes the cases where documents are assigned to distinct classes at different moments. Moreover, in that work the author proposed a visualization tool aimed at analyzing the feature space (in a binary classification setting) and at providing clues about the varying behavior of the most predictive features as time goes by. A real textual data set, composed of news articles, was characterized according to the three mentioned drifting patterns, and was shown to be a very dynamic data set.

Building on this, Mourão et al. (2008) provided a characterization of these changes (under the batch text classification setting), identifying three main temporal effects: (a) the class distribution variation, which accounts for the impact of the temporal evolution on the relative frequencies of the classes; (b) the term distribution variation, which refers to changes in the representativeness of the terms with respect to the classes as time goes by; and finally, (c) the class similarity variation, which considers how the similarity among classes, as a function of the terms that occur in their documents, changes over time. In fact, the class and the term distribution variation effects correspond to the shifting class and subclass distributions discussed in Forman (2006), respectively. Furthermore, although the class similarity variation effect is not analyzed in Forman (2006), the fickle drifting pattern is not considered in Mourão et al. (2008). As a matter of fact, the fickle drift type, a very rare event that corresponds to the change of class of a given document due to some eventual correction, is probably the most difficult case to be handled. Indeed, even the strategies discussed in Forman (2006) to handle concept drift do not handle such an issue. Hence, we focus on the three temporal effects analyzed in Mourão et al. (2008), adopting the authors' proposed nomenclature.

Specifically, building on the characterization in Mourão et al. (2008), we propose a method to enable a deeper study of the temporal effects. We propose to use a factorial experimental design to quantify to what extent each of these variations impact TC algorithms, according to data sets with distinct temporal dynamics. We also instantiate the proposed methodology using three real textual data sets and four traditional TC algorithms. In comparison with previous work, our characterization method and results contribute directly to the definition of more successful strategies to deal with and to exploit temporal effects. They also provide valuable insights into the behavior of the analyzed algorithms when dealing with changing distributions.

Experimental Workload

In this section we present the experimental workload used in our analysis. We provide a brief description of the three reference data sets (next section) as well as the four TC algorithms analyzed (section following).

Reference Data Sets

The three reference data sets considered in our study consist of sets of textual documents, each document assigned to a single class (i.e., a single label problem). For clarity purposes, throughout this paper, we refer to each class by its corresponding identifier. The class identifiers used for each data set are listed in Table 1.

The data sets employed are:

ACM-DL: A subset of the ACM Digital Library with 24,897 documents containing articles related to computer science created between 1980 and 2002. We considered only the first level of the taxonomy adopted by ACM, namely, 11 classes, which remained the same throughout the period of analysis. The distribution of the 24,897 documents among the 11 classes, in the entire time period, is presented in Figure 1a.

MEDLINE: A subset of the MedLine data set, with 861,454 documents classified into seven distinct classes related to Medicine, and created between the years of 1970 and 1985. The class distribution of the 861,454 documents during the entire time period is depicted in Figure 1b.

AG-NEWS: A collection of 835,795 news articles, classified into 11 distinct classes, that spans over 573 days. This data set presents some interesting characteristics that are typical of news data sets. For instance, some topics appear and disappear very suddenly due to periodical or ephemeral events. Moreover, there is a higher variability, w.r.t., the meaning of the terms, along with a significant class imbalance, due to the very dynamic nature of the news domain. The class distribution, spanning the whole 573-day period, is shown in Figure 1c.

These data sets potentially present distinct evolution patterns, due to their own characteristics. In particular, we expect that MEDLINE exhibits more stable behavior, in comparison to the other two data sets, because it represents a more consolidated knowledge area. Thus, we expect a tendency of newly inserted terms becoming stable along the years. In contrast, we expect more dynamic changes in AG-NEWS, a natural behavior of news data sets, which tend to present higher variability in their characteristics (e.g., variations in class distributions according to transient events, hot topics, and so on).

TABLE 1. Adopted class identifiers for each reference data set. ACM-DL corresponds 24,897 documents related to the computer science area. MedLine contains 861,454 documents related to medicine. AG-NEWS is a collection of 835,795 news articles.

ACM-DL	MEDLINE	AG-NEWS
0. General Literature	0. Aids	0. Business
1. Hardware	1. Bioethics	1. Science & Technology
2. Computer Systems Organization	2. Cancer	2. Entertainment
3. Software	3. Complementary Medicine	3. Sports
4. Data	4. History	4. United States
5. Theory of Computation	5. Space Life	5. World
6. Mathematics of Computing	6. Toxicology	6. Health
7. Information Systems		7. Top News
8. Computing Methodologies		8. Europe
9. Computer Applications		9. Italy
10. Computing Milieux		10. Top Stories

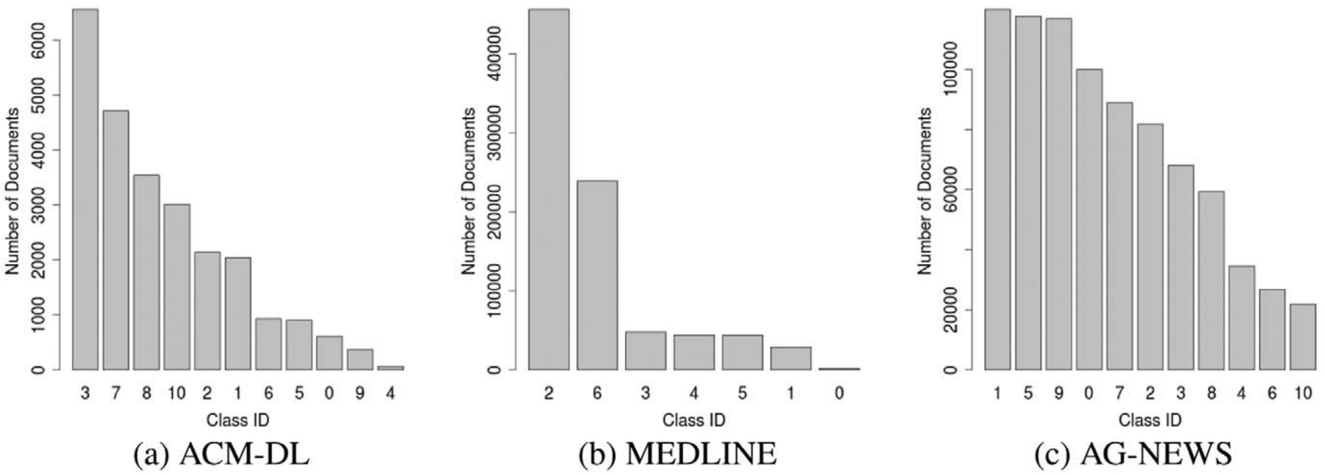
Text Classification Algorithms

We selected four representative and widely used TC algorithms to use in our characterization. These algorithms are:

Rocchio: An eager classifier that uses the centroid of a class to find boundaries between classes. The centroid of a class is defined as the average vector computed over its training examples. When classifying a new example $\vec{x}$, Rocchio associates it with the class represented by the closest centroid to $\vec{x}$.

KNN: A lazy classifier that assigns to a test document $\vec{x}$ the majority class among those of its k nearest-neighbor training documents in the vector space. Unlike Rocchio, KNN determines the decision boundary locally, considering each training document independently. We use cosine similarity to determine the nearest neighbors of a test document.

Naïve Bayes (NB): A probabilistic learning method that aims at inferring a model for each class by assigning to a test



(a) ACM-DL (b) MEDLINE (c) AG-NEWS

FIG. 1. Class distributions in the three reference data sets. In these graphs we present the total of documents per class. As we can observe, for all data sets there is an imbalance among classes in which some classes have more documents than others.

DOI: 10.1002/asi

document $\vec{x}$ the class associated with the most likely model that would have generated it. We adopt the Multinomial Naïve Bayes approach (Manning et al., 2008) because it is widely used for probabilistic text classification. The posterior class probabilities $P(\vec{x} \mid y)$ are defined as:

$$P(\vec{x} \mid y) = \eta \times P(y) \times \prod_{t \in \vec{x}} P(t \mid y), \qquad (1)$$

where η denotes a normalizing factor, $P(y)$ is the class prior probability, and $P(t|y)$ denotes the conditional probability of observing t given y. The NB classifier assigns to a test example $\vec{x}$ the class y with the highest posterior probability $P(\vec{x} \mid y)$.

Support Vector Machine (SVM): The SVM classifier aims at finding an optimal separating hyperplane between the positive and negative training documents, maximizing the distance (margin) to the closest points from either class. Given N training documents represented as pairs $(\vec{x}_i, y_i)$, where $\vec{x}_i$ is the weighted feature vector of the i^{th} training document and $y_i \in \{-1, +1\}$ the set membership of the document, SVM tries to maximize the margin between them on the training data, which leads to better classification effectiveness on test data. We may state the problem as:

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \xi_i, \text{subject to}$$

$$y_i(\vec{x}_i^T \beta + \beta_0) \geq 1 - \xi_i, \xi_i > 0,$$

where β is a vector normal to the hyperplane (the so-called weight vector), β_0 is its intercept, ξ_i is a slack variable to account for training error, C is the cost parameter, and $0 \leq i \leq N$.

We can express the SVM's decision function as:

$$\hat{F} = sign(\vec{x}^T \beta + \beta_0),$$

where the sign of the score is used to predict the example's class. Because SVM is a binary classifier, we follow the common one-against-one approach (Manning et al., 2008) to adapt binary SVM for a multiclass classification problem, as our reference data sets comprise more than two classes.

Characterization of Temporal Effects on Textual Data Sets

In this section, we briefly describe the characterization reported in Mourão et al. (2008), which uncovered three main temporal effects that affect the ACM-DL and MEDLINE data sets: (a) the class distribution variation, (b) the term distribution variation, and finally (c) the class similarity variation. More important, we also extend this prior characterization to include a third, distinct, and more dynamic data set: AG-NEWS. Our main goal is to strengthen the argument for the existence of temporal effects in the reference data sets, thus motivating our quantitative analysis of their impact on TC algorithms when applied to these data sets.

Before proceeding, we should first discretize the temporal dimension to capture the variabilities in the characteristics of the explored data sets. Time may be seen as a discretization of natural changes inherent to any knowledge area. Detectable changes, however, may occur at different time scales, depending on the characteristics of the given knowledge area. In the case of ACM-DL and MEDLINE, which are sets of scientific articles, we adopted yearly intervals for identifying such changes, as scientific conferences, for instance, usually occur once per year. For the AG-NEWS data set, we adopted, instead, a daily granularity, which should more accurately capture changes in a set of news articles. We note that, despite the differing temporal granularities observed in the data sets, we believe that this should cause no significant impact on the performed analysis. As we describe in the following sections, we analyze variations considering the most representative information in the data sets (e.g., terms with higher information gain values), ultimately discarding spurious variations that could affect the performed analyses. We eave deeper analysis regarding the impact of the data set's temporal granularity for future work. Next, we discuss the main findings of the characterization of each temporal effect in the three data sets.

Class Distribution Temporal Variation

The impact of temporal evolution on class distribution (CD) relates to the variation of the fraction of documents assigned to each class over time. CD temporal variation should be properly considered to avoid undesirable biases of the classifier used. For instance, as mentioned in the Introduction, if CD varies significantly, the "assumed" class distribution may not reflect the "true" class distribution observed when test data were created. Notice that, as an extreme case, classes may appear and disappear as a consequence of splits and joins of existing classes. For example, the subclasses *Information Retrieval* and *Artificial Intelligence* in the ACM-DL Computing Classification System (CCS) belonged to the same class—*Applications*—in 1964. Currently, each one belongs to a different class: *Information Retrieval* belongs to *Information Systems*, whereas *Artificial Intelligence* belongs to *Computing Methodologies*.

To assist the analysis of the CD temporal variation in each data set, Figure 2 shows the class probability distributions for each year of ACM-DL and MEDLINE (as in Mourão et al., 2008) and for each week of AG-NEWS.[2] The figure illustrates the variation in terms of the representativeness of the classes, that is, in terms of the fraction of document occurrences in each class, as time goes by. As the figure shows, most classes, particularly in ACM-DL and AG-NEWS, exhibit frequent oscillations in their representativeness, whereas others become less or more representative with time. For instance, the *Mathematics of Computing* class (id 6), in ACM-DL, became less representative with time, whereas the AG-NEWS *World* class (id 5) presented a

[2] We show AG-NEWS results on a weekly basis to improve readability.

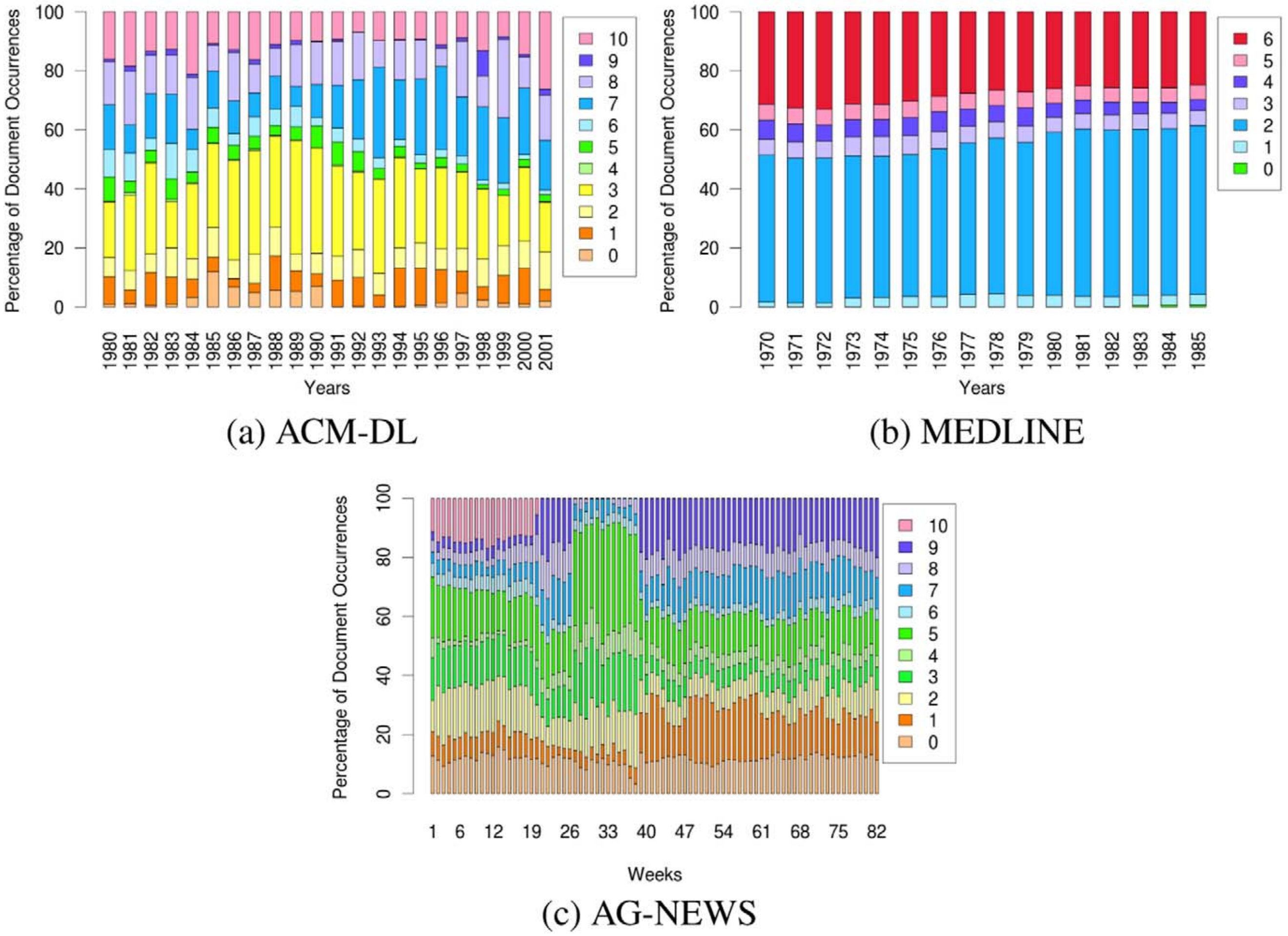


FIG. 2. Class distribution temporal variation in each reference data set. For all data sets there are frequent oscillations in their representativeness, in which the classes become less or more representative with time. This fact must be considered by classifiers to avoid generating a biased model. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

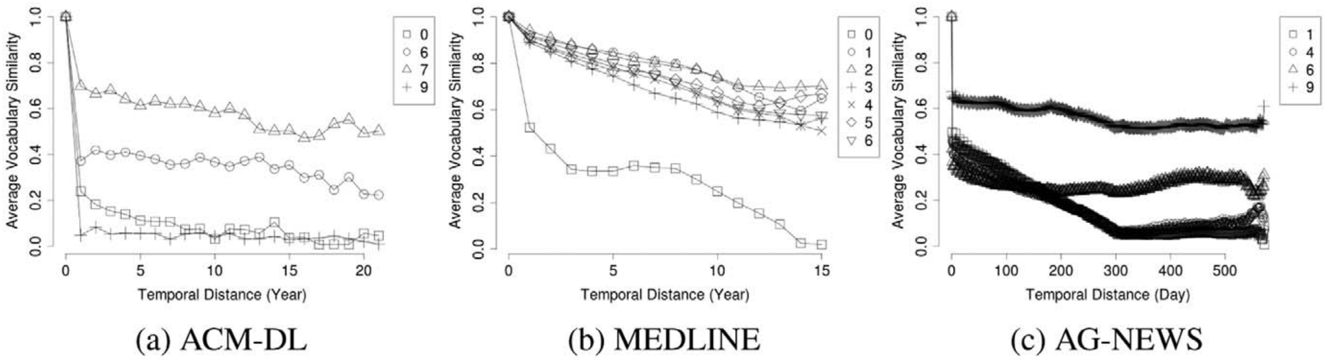
peak in its representativeness between the 25th and the 37th weeks. Another interesting case is the MEDLINE *Aids* class (id 0). Although it contains documents dating from 1970, the fraction of documents belonging to it only became significant after 1985.

These results illustrate that one needs to be very careful when creating classification models in order to avoid generating a biased model that may not be accurate for the data set to be tested. The fact that the fractions of documents in several classes are constantly changing over time, as can be seen for several classes in all three data sets in Figure 2, makes this a real problem that has be taken into account.

Term Distribution Temporal Variation

Term distribution (TD) variation is related to how the distribution of terms among the classes changes over time as a consequence of terms appearing, disappearing, and having variable discriminative power across classes. Take the following two-class example between *Mythology* and *Astrophysics*. Besides being the god of hell in Roman mythology, Pluto was also considered to be a planet until mid-2006. Up to this date, documents with the term Pluto had a higher probability of being classified in the *Astrophysics* class due to the great amount of references that mention Pluto as a planet. From this date on, since Pluto is not considered to be a planet anymore, there has been a significant reduction in the number of documents referring to it in this context. In mythology, however, the reference of Pluto remained unaltered. In this case, the term Pluto lost discriminative power in the class *Astrophysics* and gained it in the class *Mythology*. Intuitively, we may state that the TD evolution concerning the most informative terms usually happens gradually, so that the distribution of terms observed at time periods that are closer time-wise tend also to be more similar.

To characterize the TD temporal variation effect, we define, for each class and moment, the *class vocabulary* as the set of terms that have the highest values of information gain (Forman, 2003) in that moment. The vocabulary of a class at a given discretized point in time *t* represents that class in *t*. We compare the vocabularies produced for the same class across all moments using the normalized cosine similarity between them. More specifically, let $\vec{v}_{c,p_i}$ and $\vec{v}_{c,p_j}$ be the TFIDF vectors of the top *K* terms with highest information gain



(a) ACM-DL (b) MEDLINE (c) AG-NEWS

FIG. 3. Term distribution temporal variation of each reference data set. For these graphs we calculate the the average cosine similarities as we vary the time distance between the vocabularies. Note that, the class vocabularies are varying significantly over time and this variation should be considered by classifiers.

observed in class c at p_i and p_j, respectively.[3] The normalized cosine similarity between both vectors is given by:

$$sim(\vec{v}_{c,p_i}, \vec{v}_{c,p_j}) = \frac{\langle \vec{v}_{c,p_i}, \vec{v}_{c,p_j} \rangle}{\left\|\vec{v}_{c,p_i}\right\| \cdot \left\|\vec{v}_{c,p_j}\right\|},$$

where <·, ·> denotes the dot product between both vectors.

Figure 3 shows the average cosine similarities as we vary the time distance between the vocabularies. For the sake of clarity, we present results for a subset of the classes of each data set, since the same behavior is observed for all classes. Clearly, for all three data sets, the class vocabularies are varying significantly over time. For the less stable ACM-DL and AG-NEWS data sets, the similarities drop significantly even for a time distance equals to 1.[4]

Because the class vocabulary changes significantly with time, it becomes clear that a classification model generated considering documents created at a certain period of time may be less effective when tested using documents from another period of time. This is because the vocabulary may have changed in a way that the assumptions made when learning the classifier may no longer hold, that is, the discriminative terms may not be the same.

To further investigate the influence of varying class vocabularies on classification effectiveness, we perform a correlation analysis between both factors. Let $\mathbb{P}$ be the set of all points in time observed in the reference data set. For each timepoint pair $(p_1, p_2) \in \mathbb{P} \times \mathbb{P}$, a classifier h_{p_1} is learned with training examples from p_1. Test data from p_2 is then classified by h_{p_1}. To isolate the class distribution effect on classifiers (e.g., bias towards larger classes), random training data examples from p_1 are selected (with replacement) and duplicated, until equal class sizes are obtained.[5] Classification accuracy for each class c is then computed.

Let c be an arbitrary class c and $\delta = |p_1 - p_2|$. For each tuple (c, p_1, p_2), the vocabulary similarity $S_{(c,p_1,p_2)}$ and the classification accuracy $A_{(c,p_1,p_2)}$ are assessed and further summarized according to δ, through the average vocabulary similarity $S_{c,\delta}^{avg}$ and the average classification accuracy $A_{c,\delta}^{avg}$. The Pearson correlation coefficient is computed between $S_{(c,\delta)}^{avg}$ and $A_{(c,\delta)}^{avg}$, together with a 99% confidence interval.

In Figure 4, we report the correlations obtained using the Naïve Bayes classifier. The analysis shows that, considering the Naïve Bayes classifier, there is a positive correlation between vocabulary similarity and classification accuracy, for all classes but one (class *Cancer* in Medline, whose correlation coefficient is not statistically superior to zero—even though with positive mean value).[6]

As shown in Figure 3, as $\delta = |p_1 - p_2|$ increases, the vocabulary similarity between p_1 and p_2 decreases. This is observed for all classes. Due to the observed positive correlation, as the vocabulary similarity decreases, so does the classification accuracy. This is an expected result and is in accordance with our assumption that variations in the class similarity as time goes by hurt classification effectiveness.

Class Similarity Temporal Variation

Finally, class similarity (CS) variation relates to how the pairwise similarity among classes, as a function of the terms that occur in their documents, varies over time. The similarity between two arbitrary classes may change over time due to the migration and variation of the frequency of the terms in their vocabularies: two classes may be similar at a given moment, and become less similar later in the future, and vice versa.

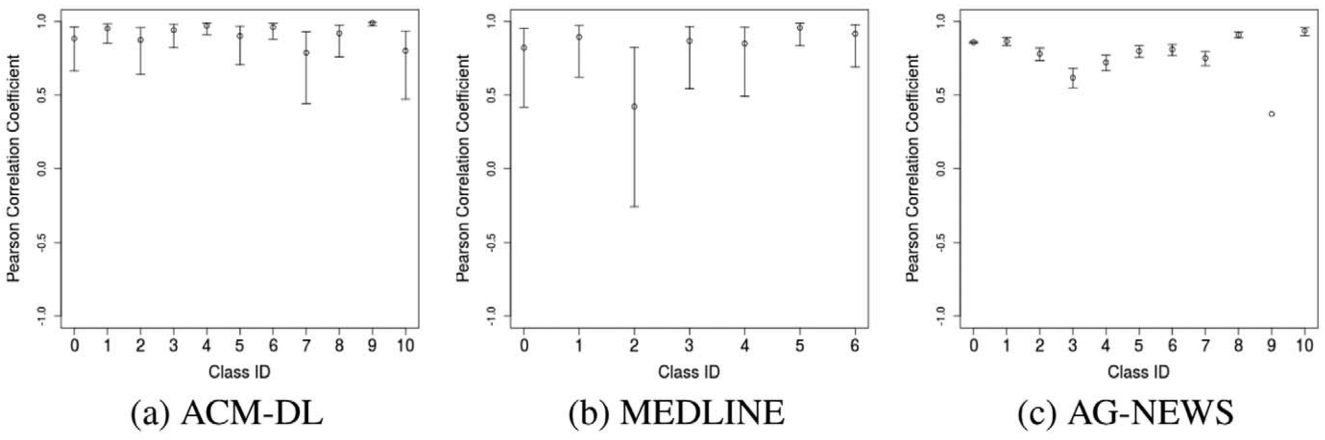
To analyze the CS temporal variation, we calculate the cosine similarity between the vocabularies (given by the TFIDF vectors of the top K terms with highest information gain) of each pair of distinct classes at every point in time

[3] The IDF of each term is computed considering only the documents created at the specified moment.

[4] At time distance zero, the similarity, for all classes, is equal to 1, because we are comparing a vocabulary to itself, which obviously corresponds to the maximum possible similarity value.

[5] We duplicate data to avoid generating new information generated—which could influence our analysis in terms of accuracy.

[6] The same correlation analysis was applied to the remaining classifiers (i.e., Rocchio, KNN, and SVM). Similar results were observed and can be found in the online appendix.



(a) ACM-DL

(b) MEDLINE

(c) AG-NEWS

FIG. 4. Pearson correlation coefficient computed for each class vocabulary similarity and classification effectiveness pair, as we vary the time distance between the vocabularies and training/test data—Naïve Bayes. Positive correlations indicate that, as the vocabulary similarity decreases, so does the classification accuracy.

(years for ACM-DL and MEDLINE, and days for AG-NEWS). Table 2(a), 2(b), and 2(c) show the results for MEDLINE, ACM-DL, and AG-NEWS, respectively. Each entry in the tables comprises the coefficient of variation (i.e., the ratio of the standard deviation to the mean) of the similarities between the associated pairs of classes computed over all moments.[7]

As we can observe, the similarities between some pairs of classes vary significantly with time. For example, the similarities between *General Literature* (id 0) and *Computer Systems Organization* (id 2) in ACM-DL, *Aids* (id 0) and *Cancer* (id 2) in MEDLINE, and *Italy* and *Top Stories* (ids 9 and 10, respectively) in AG-NEWS have coefficients of variation equal to 1.79, 0.77, and 2.40, respectively. This means that these pairs of classes may have been very similar in some periods, but also loosely related in others. Thus, the difficulty in separating them varies significantly as time passes.

Summarizing this discussion, there is clear evidence of temporal variations in the class and term distributions as well as on the similarities among classes in all three analyzed data sets. These variations may ultimately affect the performance of classifiers. In the next section we detail the proposed method to quantify the impact of each of these three temporal effects on the effectiveness of TC algorithms.

Experimental Design

In this section we describe our method for assessing the impact of the identified temporal effects on each TC algorithm and textual data set. The core component of our method is a factorial experimental design (Jain, 1991). This technique has already been applied in multiple contexts to quantify the effect of different factors and interfactor interactions on a given response variable (see examples in de Lima, Pappa, de Almeida, Gonçalves, & Meira, 2010; Jain, 1991; Orair, Teixeira, Wang, Meira, & Parthasarathy, 2010; Vaz de Melo, da Cunha, Almeida, Loureiro, & Mini, 2008). However, to the best of our knowledge, this is the first time it is applied to assess the impact of temporal effects on TC algorithms. As will be discussed below, the application of this technique in this context is challenging in itself.

We start by reviewing the factorial design procedure in general terms in the next section. We then discuss how it may be applied to evaluate the impact of temporal effects on TC algorithms in the following section, and present its application to the four selected TC algorithms and the three chosen textual data sets in the following section.

Factorial Design

The factorial experimental design, jointly with the simple design, where each factor is varied independently, are the most widely used experimental methodologies (according to Jain, 1991). It is a general-purpose method that has been applied to a large variety of different domains, such as social science, agriculture, and psychology (Box, Hunter, & Hunter, 1978; Shuttleworth, 2014). For example, social researchers often use factorial designs to assess the effects of educational methods, while taking the influence of socioeconomic factors and background into account. Similarly, agricultural science often uses factorial designs to assess the impact of particular variables on crops. In computer science, factorial design has been applied to diverse problems such as studying the impact of parameters on evolutionary algorithms (de Lima et al., 2010), as well as evaluating the cooperation among different wireless sensor networks (Vaz de Melo et al., 2008).

The goal of a factorial design is to quantify the impact of different factors on a response variable. Unlike simple designs, it covers all possible combinations of factor levels, and thus allows us to quantify the impact of the interfactor

[7] Note that, in contrast to Mourão et al. (2008), we here characterize class similarity variation using the coefficient of variation metric, instead of the standard deviation of the pooled similarities, as the latter depends on the unit and scale of the measurements.

TABLE 2. Coefficients of variation of the pooled pairwise class similarities for each reference data set. In this case, we calculate the cosine similarity between the vocabularies between each pair of classes over all moments. The similarities between some pairs of classes vary significantly with time and also should be considered by the classifiers.

(a) ACM-DL

Class ID	0	1	2	3	4	5	6	7	8	9	10
0	0.00	1.25	1.79	0.97	1.71	1.13	1.14	1.50	0.91	0.93	0.80
1	—	0.00	0.40	0.36	0.86	0.38	0.39	0.43	0.28	0.80	0.48
2	—	—	0.00	0.30	0.89	0.63	0.50	0.33	0.27	0.86	0.34
3	—	—	—	0.00	0.72	0.46	0.37	0.27	0.41	0.77	0.34
4	—	—	—	—	0.00	0.90	0.92	0.93	0.80	1.32	1.00
5	—	—	—	—	—	0.00	0.48	0.52	0.62	1.20	0.75
6	—	—	—	—	—	—	0.00	0.48	0.36	1.33	0.48
7	—	—	—	—	—	—	—	0.00	0.39	0.84	0.45
8	—	—	—	—	—	—	—	—	0.00	0.68	0.34
9	—	—	—	—	—	—	—	—	—	0.00	0.73
10	—	—	—	—	—	—	—	—	—	—	0.00

(b) MEDLINE

Class ID	0	1	2	3	4	5	6
0	0.00	0.57	0.77	0.65	0.54	0.67	0.67
1	—	0.00	0.20	0.31	0.30	0.17	0.18
2	—	—	0.00	0.07	0.22	0.12	0.04
3	—	—	—	0.00	0.32	0.07	0.06
4	—	—	—	—	0.00	0.15	0.20
5	—	—	—	—	—	0.00	0.05
6	—	—	—	—	—	—	0.00

(c) AG-NEWS

Class ID	0	1	2	3	4	5	6	7	8	9	10
0	0.00	0.56	0.52	0.72	0.67	0.47	0.75	0.56	0.63	0.95	1.92
1	—	0.00	0.64	0.85	0.87	0.78	0.80	0.74	0.81	1.43	1.98
2	—	—	0.00	0.64	0.68	0.50	0.71	0.61	0.66	1.27	1.84
3	—	—	—	0.00	0.81	0.70	0.87	0.77	0.78	0.98	1.95
4	—	—	—	—	0.00	0.67	0.74	0.67	0.74	1.17	2.08
5	—	—	—	—	—	0.00	0.69	0.44	0.58	1.33	1.84
6	—	—	—	—	—	—	0.00	0.79	0.77	1.40	2.06
7	—	—	—	—	—	—	—	0.00	0.62	1.15	2.12
8	—	—	—	—	—	—	—	—	0.00	1.26	2.06
9	—	—	—	—	—	—	—	—	—	0.00	2.40
10	—	—	—	—	—	—	—	—	—	—	0.00

interactions, which might be relevant. In other words, it is particularly interesting in scenarios where one expects that some relevant interactions might exist or want to assess whether they exist. This is exactly the scenario of our study, as we want to quantify the impact of specific factors (temporal effects) and their interactions on the effectiveness of TC algorithms.

In more detail, given k factors (the so-called independent variables), which may assume n levels (possible values), and a response variable, a full factorial n^k experimental design aims at quantifying the impact of each individual factor as well as all interfactor interactions (of all orders) on the given response variable. In other words, it aims at quantifying the *effect* of these factors and interactions on the variations observed in the response across a series of n^k experiments, carefully designed to cover all possible configurations of factor levels.

To conduct the n^k design, the parameters that affect the system under study should be carefully controlled to avoid misleading conclusions due to unexpected effects. Thus, one has to be able to isolate and carefully vary the *factors*, which are parameters related to the goals of the study and thus selected to be analyzed, while controlling the other parameters, which are kept fixed. Usually, factors are varied from smaller to larger values, based on the assumption of monotonicity, that is, the response variable continuously increases (or decreases) as the factor value becomes larger.

In many scenarios, the system under study presents an inherent variability, and, thus, measurements are susceptible to inaccuracies, referred to as *experimental errors*. In such cases, the impact of both the factors and their interactions should be assessed in comparison to such errors, and an experimental design with r replications ($n^k r$) should be adopted. This is done by replicating the measurements for each factor-level combination r times. It is important to emphasize the need for controlling all parameters with significant impact on the system, by either treating them as factors or keeping them fixed, as the effect of uncontrolled parameters that cannot be distinguished from experimental errors.

Such an experimental design is typically used as a primary tool to help one sort factors and interfactor interactions in terms of their impact on the response variable, thus providing quantitative evidence of which factors (and/or interactions) are more relevant for further investigation. The examination of every possible factor-level combination enables one to have a complete picture of the system behavior regarding the factors considered. However, it comes at the expense of a potentially very costly study. The required number of experiments (i.e., $n^k r$ experiments) may be too large and unfeasible to perform due to resource and time constraints. One of the most recommended strategies to reduce the number of required experiments consists of reducing the number of levels considered for each factor (Jain, 1991). As a matter of fact, for an initial assessment, one can consider only two levels (lower and upper levels) of each factor, thus performing a $2^k r$ factorial design. By doing so, one can determine the relative importance of all factors and interactions, and leave for a more detailed study the analysis of more levels of the most relevant factors.[8]

We describe the main steps of a $2^k r$ factorial design using, for illustration purposes, $k = 2$ factors, referred to as A and B. The $2^2 r$ design aims to fit an additive model that characterizes the impact of each factor A and B as well as of its interaction AB on the response variable y. This model is given by:

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B + \varepsilon, \qquad (2)$$

where q_0 is the mean value of the response variable, q_A, q_B, and q_{AB} stand for the effects associated with factors A, B and interaction AB, and ε denotes the experimental errors. For each factor $f \in \{A, B\}$, a variable x_f is defined as:

$$x_f = \begin{cases} -1 & \text{if } f \text{ is at the lower level,} \\ +1 & \text{if } f \text{ is at the upper level.} \end{cases}$$

Thus, q_f denotes the extent of the variation on the global average q_0 imposed by factor f, on average.

The $2^2 r$ experimental design can be summarized into five steps. Step 1 consists of parameter estimation, in which we compute q_0 and the effects q_A, q_B, and q_{AB}. Once the effects have been computed, the model can be used to estimate the response for any given factor values (x-values). For instance, the estimated response when factors A and B are at levels x_{Ai} and x_{Bi}, respectively, is computed as:

$$\hat{y}_i = q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{Ai} x_{Bi}. \qquad (3)$$

The importance of a factor can be measured by the proportion of the total variation in the response variable that can be explained by it. Thus, in Step 2 we compute the variation of response y across all experiments that can be explained by each factor (SS_A, SS_B, SS_{AB}, respectively) as well as the variation that remains unexplained, being thus credited to experimental errors (SS_E). In other words, we compute $SS_f = 2^k r q_f^2 \quad (f \in \{A,\ B,\ AB\})$, and $SS_E = \sum_{i=1}^{2^k} \sum_{j=1}^{r} e_{ij}^2$, where error e_{ij} denotes the difference between the estimated response for the i^{th} experiment ($\hat{y}_i$) and the value measured in its j^{th} replication (y_{ij}). The total variation, referred to as the Sum of Squares Total (SS_T), is also computed as the sum of SS_A, SS_B, SS_{AB} and SS_E.

Next (Step 3), we express the SS_f ($f \in \{A, B, AB\}$) and SS_E as percentages of the total variation SS_T, so as to more easily assess the importance of each factor and of the experimental errors in the observed response variations. Factors (and interactions) that explain a higher percentage of the total variation are considered more important and, thus, are candidates for further analysis.

Because the effects are computed from a sample, they are indeed random variables, and could take different values if another set of experiments was performed. Thus, it is necessary to compute their associated confidence intervals (Step 4). We do so by first computing the root mean square of errors (*RMSE*) and the standard deviation s_f of each effect q_f ($f \in \{A, B, AB\}$). *RMSE* denotes the standard error of the estimates, thus measuring how well the model explains the observations. It is computed as the square root of the ratio of SS_E to the degrees of freedom associated with the experimental errors (in the current design, $2^2(r-1)$).[9] The $100(1-\alpha)\%$ two-sided confidence intervals are computed using either a Student's t distribution or z distribution, depending on the degrees of freedom $2^2(r-1)$ (see Jain, 1991). Any effect whose confidence interval does not include zero is statistically significant with the given confidence.

Finally, in Step 5, we assess the model quality, by means of the coefficient of determination R^2. This is done by comparing the unexplained variation (SS_E) with the total variation (SS_T), being a measure of goodness of the fit for the additive model in Equation 2. The closer to 1, the better the fitted model.

In performing the factorial design, we assume that the model errors are additive, statistically independent, and normally distributed with zero mean and constant standard deviation. We also assume that the effects of factors are additive. It is important to check if any of these assumptions are violated because, if so, the conclusions based on the estimated model would be misleading. Jain (1991) presents some visual tests that may be applied in order to validate these assumptions.

The general procedure to perform a $2^k r$ design, for any values of k and r, is presented in Algorithm 1.

[8] Even though there might be other methods that could be adopted in our target problem, we argue that factorial design (and particularly $2^k r$ design) provides a suitable and cost-effective tool to achieve our proposed goals.

[9] In a general $2^k r$ design, the degrees of freedom of the experimental errors is given by $2^k(r-1)$.

Algorithm 1. Factorial design procedure.

```
function FACTORIAL DESIGN
Step 1: Estimate model parameters (i.e., grand mean and factor effects)
    $q_0 \leftarrow \frac{1}{2^k r}\sum_{i=1}^{2^k}\sum_{j=1}^{r} y_{ij}$
    $q_f \leftarrow \frac{1}{2^k}\sum_{i=1}^{2^k} x_{fi}\hat{y}_i$, where $f \in [1, 2^k - 1]$ and $\hat{y}_i = \frac{1}{r}\sum_{j=1}^{r} y_{ij}$
Step 2: Compute total variation and variation due to each factor and experimental errors
    $SS_f \leftarrow 2^k r q_f^2$, where $f \in [1, 2^k - 1]$
    $SS_E \leftarrow \sum_{i=1}^{2^k}\sum_{j=1}^{r} e_{ij}^2$, where $e_{ij} = y_{ij} - \hat{y}_i$
    $SS_T \leftarrow \sum_{f=1}^{2^k-1} SS_f + SS_E$
Step 3: Compute percentage of variation each factor/error is responsible for
    $P_f \leftarrow \frac{SS_f}{SS_T} \times 100$, where $f \in [1, 2^k - 1]$
    $P_E \leftarrow \frac{SS_E}{SS_T} \times 100$
Step 4: Compute confidence intervals of the effects
    $RMSE \leftarrow \sqrt{\frac{SS_E}{2^k(r-1)}}$
    $s_f \leftarrow \frac{RMSE}{\sqrt{2^k r}}$, where $f \in [1, 2^k - 1]$
    $CI_f \leftarrow q_f \pm t_{\left[1-\frac{\alpha}{2};2^k(r-1)\right]} s_f$, where $f \in [1, 2^k - 1]$
Step 5: Assess model accuracy by the coefficient of determination
    $R^2 \leftarrow 1 - \frac{SS_E}{SS_T}$
```

Applying 2^kr *Design in the Characterization of Temporal Effects*

In this section we describe how the $2^k r$ design can be applied to quantify the impact of temporal aspects on TC algorithms, considering different data sets. As we focus on three different temporal aspects, namely, the class distribution temporal variation (*CD*), the term distribution temporal variation (*TD*), and the class similarity temporal variation (*CS*), our experimental design takes $k = 3$ factors. The two levels considered for each factor, which we call "lower" and "upper" levels, defined next, refer thus to the degree of temporal variation observed on it. Given a reference data set and a TC algorithm, the goal is to partition the document set into 2^3 groups corresponding to all possible factor-level configurations, and then evaluate the algorithm for each configuration, considering the grouped documents. We then apply the $2^k r$ design procedure, described in Algorithm 1, to quantify the effect of each factor and interfactor interaction on the effectiveness of the TC algorithm.

The response variable y is thus the classification effectiveness, which is here assessed by the commonly used F_1 measure. F_1, the harmonic mean between the precision p and the recall r, is given by:

$$F_1 = \frac{2pr}{p+r}$$

where precision p is the percentage of documents assigned by the classifier to class c_i that were correctly classified, and recall r is the percentage of documents belonging to class c_i that were correctly classified.[10]

[10] The described F_1 measure corresponds to the overall performance of the methods across all classes. Using a per-class variation of the measure (also known as Macro-F_1) would imply having to consider another parameter in the analysis: the class imbalance. To focus our analysis on the time-related factors, the goal of the present study, we would have to isolate or control this parameter. However, possible approaches to isolate this parameter (e.g., under- or oversampling; Lin, Hao, Yang, & Liu, 2009; Liu, Ghosh, & Martin, 2007) are typically very hard to perform in practice

For each configuration, we run a number r of replications following a cross-validation strategy, commonly adopted by the machine learning community. There are, at least, two usual approaches for doing so: the K-fold cross-validation and the repeated random subsampling. The K-fold cross-validation consists of randomly splitting the data into K independent folds. At each iteration, one fold is retained as the test set, and the remaining $K - 1$ folds are used as the training set. The repeated random subsampling consists of randomly selecting a fraction of documents from the data set, without replacement, to compose the test set, and the remaining documents retained as the training set. This is performed for each replication. Because in the K-fold cross-validation the size of the folds depend on the number of iterations, it becomes more suitable to medium/large-sized data sets, whereas the repeated random subsampling is usually adopted for small-sized data sets when the number of replications is large. We discuss the technique adopted for each analyzed data set below.

One challenge to build our factorial design is how to define the 2^3 groups of documents. To that end, we must quantify the temporal variation of each factor in the set of documents of the reference data set, define the two levels for each factor and, based on them, group the documents according to all possible factor-level combinations. The following three sections describe how we performed these steps for the *CD*, *TD*, and *CS* factors. Note that, because *CD* and *CS* relate exclusively to the characteristics of the class to which a document belongs, we define the *CD* and *CS* levels associated with a document based on the corresponding values of its class. *TD*, on the other hand, relates to the relationships among terms and classes. Thus, to define the *TD* level associated with a document, isolating this factor from the others, we adopt a finer grained approach that analyzes the document's contents. After defining the factor levels, we discuss a few other aspects that require attention to avoid misleading results.

Class distribution: lower and upper levels. Let $\mathbb{C}$ and $\mathbb{P}$ be the set of classes and points in time observed in the reference data set, respectively. To isolate the class distribution effect into lower and upper levels, we consider the relative sizes of the classes (i.e., fraction of the data set documents assigned to the classes) at each point in time $p \in \mathbb{P}$. For each class $c \in \mathbb{C}$, we compute the coefficient of variation ν (i.e., the ratio of the standard deviation to the mean) of the relative size of c for all values of p. The coefficient of variation is used since it is dimensionless and scale invariant, and thus more appropriate to deal with temporal changes in class distribution observed in the reference data set.

We then partition the documents into two groups based on a given threshold δ_{CD}: those whose classes present ν values lower than δ_{CD} are assigned to the "lower" group ($CD\downarrow$, with associated variable $x_{CD} = -1$), and those whose classes present ν values higher than δ_{CD} are assigned to the "upper" group ($CD\uparrow$, with associated variable $x_{CD} = +1$). We define the δ_{CD} threshold as the median of all computed coefficients of variation.

Term distribution: lower and upper levels. We determine the *TD* level to which a document belongs computing the *document stability level*, which is characterized by the density of the documents' terms that are stable. To assess the stability of a given term, we use the concept of *stability period* (Rocha et al., 2008).

[Stability period] Let $DF(t, c, p)$ be the number of documents belonging to class $c \in \mathbb{C}$ that contain term t and were created at the point in time $p \in \mathbb{P}$. A stability period S_{t,p_r} of a term t considering $p_r \in \mathbb{P}$ as the reference point in time, is the set of points in p that compose the largest continuous period of time,[11] starting from p_r and growing both to the past and the future, until there exists some class c such that:

$$dominance(t, c, p) = \frac{DF(t, c, p)}{\sum_{c' \in \mathbb{C}} DF(t, c', p)} > \alpha,$$

for some predefined $0 \leq \alpha \leq 1$.

We characterize the stability of a term t, regarding a reference point in time p_r, by the *term stability level* (TSL), defined as:

$$TSL(t, p_r) = \frac{|StabilityPeriod(t, p_r)|}{|\mathbb{P}|}$$

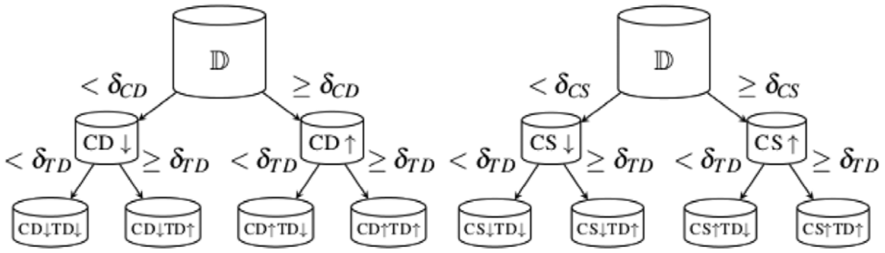
We then use the TSL to estimate the document stability level (DSL) of a given document d. Let p be the point in time when d was created. We define the DSL of d as:

$$DSL(d) = \frac{\sum_{t \in d} TSL(t, p)}{|\{t \mid t \in d\}|}$$

As we can observe, $0 \leq DSL(d) \leq 1$, where the lower bound ($DSL(d) = 0$) occurs for documents without stable terms, and the upper bound ($DSL(d) = 1$) occurs for documents composed only by terms t with maximal $TSL(t, p_r)$, that is, terms that have stability periods with maximum duration regarding the time when d was created.

without affecting the temporal factors, which ultimately could compromise our study. Thus, we leave for future work the consideration of Macro-F1 in our experimental design. We note, however, that, in the absence of very skewed class distributions, both variations of the F_1 metric tend to produce compatible results.

[11] We consider the same definition of stability period as Rocha et al. (2008), adopting a *continuous* period of time, due to computational feasibility. Considering noncontinuous intervals increases the search space exponentially with the number of points in time ($2^{|\mathbb{P}|}$ possible intervals to be considered). This is a safe decision because, as we can see in Figure 3, the variations observed in the relationships between terms and classes are smooth, that is, we do not observe any abrupt steps in the curves.



(a) $CD{\times}TD$ Partitioning.

(b) $CS{\times}TD$ Partitioning.

FIG. 5. Document partitioning over the 2^2 cells in the $CD \times TD$ and $CS \times TD$ experimental designs. All documents of each data set are divided into four partitions, with the same number of documents randomly sampled at each replication, thus covering all possible factor-level combinations: {$CD{\downarrow}TD{\downarrow}$, $CD{\downarrow}TD{\uparrow}$, $CD{\uparrow}TD{\downarrow}$, $CD{\uparrow}TD{\uparrow}$} for the former and {$CS{\downarrow}TD{\downarrow}$, $CS{\downarrow}TD{\uparrow}$, $CS{\uparrow}TD{\downarrow}$, $CS{\uparrow}TD{\uparrow}$} for the latter, where ↓ and ↑ denote "lower" and "upper" levels, respectively.

The documents are then partitioned into two groups: those with DSL less than a predefined threshold δ_{TD} are assigned to the "lower" group ($TD{\downarrow}$, with associated variable $x_{TD} = -1$), and the remaining documents are assigned to the "upper" group ($TD{\uparrow}$, with associated variable $x_{TD} = +1$). We define the δ_{TD} threshold as the median DSL value computed over all documents.

Class similarity: lower and upper levels. The "lower" group ($CS{\downarrow}$, with associated variable $x_{CS} = -1$) is composed of documents whose classes are more stable in terms of their similarities with other classes during the whole period covered in the reference data set. Accordingly, the "upper" group ($CS{\uparrow}$, with associated variable $x_{CS} = +1$) is composed of documents whose classes present higher variability in their similarities with other classes. To quantify this variability for a class c, we first compute the similarity $sim(\vec{v}_{c,p}, \vec{v}_{c',p})$ where $c,\ c' \in \mathbb{C},\ c \neq c'$ and $\vec{v}_{c,p}$ denotes the vocabulary (TFIDF vector) of class c at the point in time $p \in \mathbb{P}$. The vocabulary of class c at time p consists of the top K terms with highest information gain (Forman, 2003) in c at that time. We then compute the coefficient of variation ν of the $(|\mathbb{C}| - 1)|\mathbb{P}|$ pooled similarities between class c and all other classes in $\mathbb{C}$, in all points in time in $\mathbb{P}$. We separate documents into two groups based on the ν values of their classes and on a predefined threshold δ_{CS} which, similar to the other thresholds, is defined as the median coefficient of variation computed across all classes.

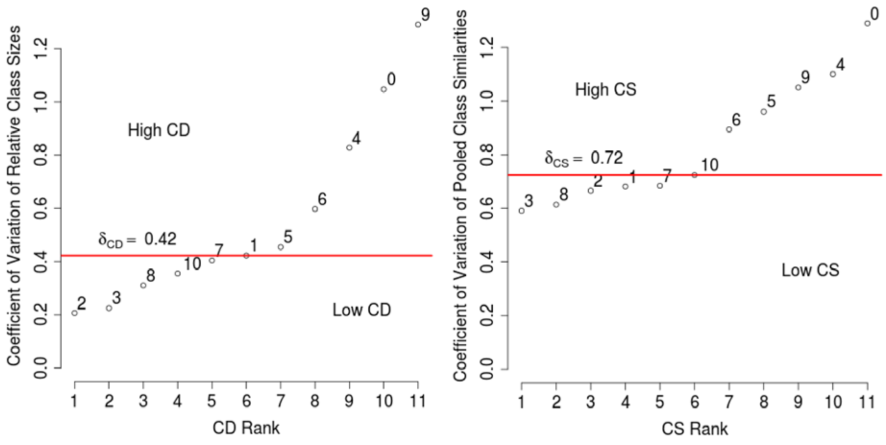
Other challenging aspects. As a requirement of a careful experimental design, we have to control the parameters that may influence the responses but are not the target of the analysis (i.e., are not treated as factors in the design). One such parameter is the *learning curve* effect, characterized by the differences in classification effectiveness obtained by varying the size of the training set. As is well known, as the training set used by supervised learning strategies becomes larger, the more information is available to build the classification model, which ultimately influences the effectiveness of the classifier. If we neglect such matters, and consider different training set sizes for each factor-level combination, we may mask the actual impact of the temporal effects on the TC algorithms. Clearly, we need to isolate the learning curve effect to remove its influence on the response variable. Therefore, for each experimental replication we randomly selected the same number of documents for each of the 2^k partitions, according to the size of the smaller partition. This ensures training sets with equal sizes across all factor-level combinations, thus isolating such an effect.

Another important data set-dependent aspect is that the documents and classes in the reference data set must fulfill all 2^3 groups to enable us to conduct the proposed experimental design. However, in some cases, as in the reference data sets analyzed here, this might not hold, particularly due to combinations regarding the CD and CS factors (see discussion below). In such cases, we are not able to isolate and simultaneously analyze all three temporal factors. To overcome this issue, and yet provide valuable insights about the temporal effects, we propose a pairwise approach, consisting of two 2^2r designs, referred to as $CD \times TD$ and $CS \times TD$ designs. This decision comes with a cost, as we are not able to analyze a possible interaction between CD and CS. However, as we see in the next section, these two factors are typically highly correlated. Thus, analyzing them in separate experimental designs might still be worthwhile.

The first experimental design, $CD \times TD$, aims at analyzing the impact of CD, TD, and their interaction on the classification effectiveness achieved by the four algorithms in the three reference data sets. The second one, referred to as $CS \times TD$, allows one to quantify the impact of CS, TD, and their interaction. For both designs, all documents of each reference data set are divided into four partitions, with the same number of documents randomly sampled at each replication, covering thus all possible factor-level combinations: {$CD{\downarrow}TD{\downarrow}$, $CD{\downarrow}TD{\uparrow}$, $CD{\uparrow}TD{\downarrow}$, $CD{\uparrow}TD{\uparrow}$} for the former and {$CS{\downarrow}TD{\downarrow}$, $CS{\downarrow}TD{\uparrow}$, $CS{\uparrow}TD{\downarrow}$, $CS{\uparrow}TD{\uparrow}$} for the latter, where ↓ and ↑ denote "lower" and "upper" levels, respectively. Such a partitioning scheme is illustrated in Figure 5.

Finally, as mentioned above, it is necessary to verify the assumptions that the model errors are independent and

DOI: 10.1002/asi



(a) Class Distribution Variation (*CD*)

(b) Class Similarity Variation (*CS*)

FIG. 6. Determining the lower and upper levels of *CD* and *CS*—ACM-DL. Note there is a high correlation between these two factors, which supports our decision to ignore a possible interaction between them, decoupling the analysis into two separate 2^2r factorial designs. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

normally distributed with zero mean and constant standard deviation, and that the effects of factors are additive. We performed the visual tests discussed in Jain (1991) for all factorial designs reported in the following sections, and all these assumptions were properly validated. The performed visual tests can be found in the online appendix.[12] Furthermore, we also performed the factorial designs under the log transformed response space to contrast the multiplicative model and the additive one. As expected, both models were consistent in terms of allocation of variance among the effects and interactions.

Quantifying the Impact of Temporal Effects on TC

In this section we describe how we applied the proposed method to quantify the impact of temporal effects on TC using, as experimental workload, the four TC algorithms (Rocchio, KNN, Naïve Bayes, and SVM) and the three textual data sets (ACM-DL, MEDLINE, and AG-NEWS) presented previously. In other words, we performed a series of experiments, following the proposed method, for each combination of TC algorithm and reference data set. As the number of available documents in all three data sets are not enough to cover all 2^3 partitions, we adopted the strategy described above, conducting two separate 2^2r designs in each case.

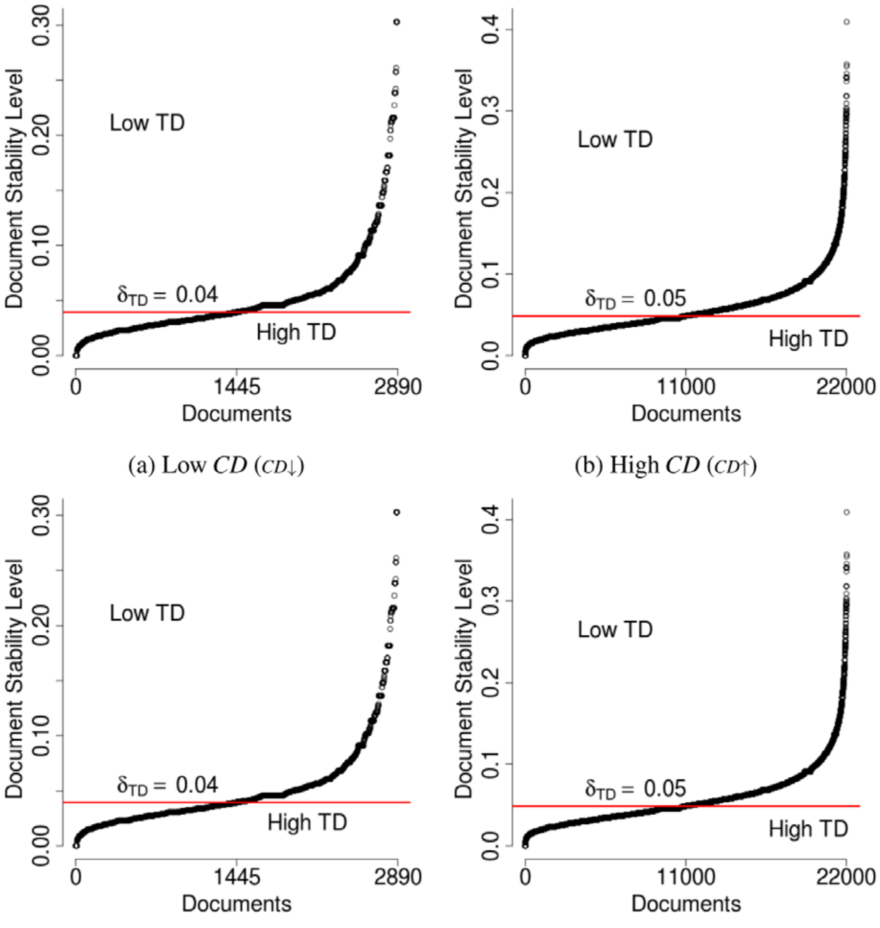
Recall that, in "Term Distribution, Lower and Upper Levels" (previously), to define the $TD\downarrow$ and $TD\uparrow$ document groups, we must determine the dominance threshold α to compute the stability periods. Different values of α were evaluated and, as they lead to similar results, we fixed $\alpha = 50\%$, ensuring that the terms will have a high degree of exclusivity with a single class. Furthermore, as described previously, both KNN and SVM classifiers have some tuning parameters. In particular, one must define the number of nearest neighbors to be considered (parameter k) to use KNN. Considering the SVM classifier, we used the LIBLINEAR library, an efficient linear SVM implementation (Fan, Chang, Hsieh, Wang, & Lin, 2008) well suited for text classification. For such a classifier, the tuning parameter is the cost C of training misclassification. We employed a one-against-one procedure to adapt the binary SVM to the multiclass scenario because this is the case in our reference data sets. All parameters were calibrated through cross-validation performed in the training set.

Next, we discuss the experimental design conducted for each reference data set.

ACM-DL. The first step is to partition the ACM-DL documents into four groups for the $CD \times TD$ design and four other groups for the $CS \times TD$ design. We do so by first partitioning them into one pair of groups for each design using the δ_{CD} and δ_{CS} thresholds, as discussed previously. The coefficient of variation values of the individual classes, along with the respective thresholds, are shown in Figure 6.

Analyzing Figure 6, we can further understand why the ideal 2^3 design was not possible to be conducted on the ACM-DL data set. Let $\mathbb{C}_{CD\uparrow}$, $\mathbb{C}_{CD\downarrow}$, $\mathbb{C}_{CS\uparrow}$, $\mathbb{C}_{CS\downarrow}$ denote the sets of classes in partitions $CD\uparrow$, $CD\downarrow$, $CD\uparrow$ and $CD\downarrow$, respectively. As we can observe, $|\mathbb{C}_{CD\uparrow} \cap \mathbb{C}_{CS\downarrow}| = 1$ and $|\mathbb{C}_{CD\downarrow} \cap \mathbb{C}_{CS\uparrow}| = 1$. As we need at least two classes in each partition to proceed with the classification task, there are not enough documents to fill all the cells of the ideal 2^3r design. Figure 6 also shows that five of the six classes with high *CS* also present high *CD* and four of five classes with low *CS* also have low *CD*. In other words, there is a high correlation between these two factors, which supports our decision to ignore a possible interaction between them, decoupling the analysis into two separate 2^2r factorial designs.

[12] The online appendix can be found at: http://www.dcc.ufmg.br/~tsalles/aq/.



(a) Low CD ($CD\downarrow$)

(b) High CD ($CD\uparrow$)

(c) Low CS ($CS\downarrow$)

(d) High CS ($CS\uparrow$)

FIG. 7. Determining the lower and upper levels of TD—ACM-DL. Documents from the $CD\downarrow$ (or $CS\downarrow$) partition with DSL smaller than the corresponding δ_{TD} are assigned to $CD\downarrow TD\downarrow$ (or $CS\downarrow TD\downarrow$) group, whereas those with DSL higher than δ_T are assigned to $CD\downarrow TD\uparrow$ (or $CS\downarrow TD\uparrow$) group. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

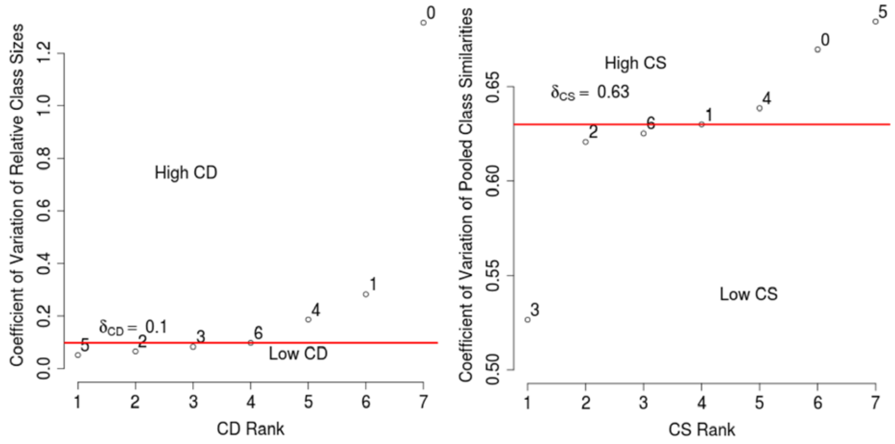
Next, we further subdivide each CD-based document partition according to the TD factor using the δ_{TD} threshold (above). We do the same for the CS-based document partitions. Figure 7 shows the distribution of document stability level (DSL) values and the δ_{TD} for each partition. Documents from the $CD\downarrow$ (or $CS\downarrow$) partition with DSL smaller than the corresponding δ_{TD} are assigned to $CD\downarrow TD\downarrow$ (or $CS\downarrow TD\downarrow$) group, whereas those with DSL higher than δ_T are assigned to $CD\downarrow TD\uparrow$ (or $CS\downarrow TD\uparrow$) group. The same applies for those documents from $CD\uparrow$ and $CS\uparrow$.

Recall that a $2^k r$ design requires r replications to be performed for each configuration and, as discussed in the previous section, this can be achieved by employing either K-fold cross-validation or repeated random subsampling. Due to the small size of the ACM-DL data set and the use of sampling to isolate the sampling effect, we use the repeated random subsampling strategy, selecting 50% of documents to compose the test set and the retaining the remaining ones the training set. We performed $r = 100$ replications.

Table 3 shows the results of both factorial designs ($CD \times TD$ and $CS \times TD$) for each TC algorithm (first column). For better presentation, we represent CD (CS) as A and TD as B for the $CD \times TD$ ($CS \times TD$) design. For each algorithm and design, the "%Var" row lists the percentage of variation in classification effectiveness that can be explained by each effect q_f ($f \in \{A, B, AB\}$) and by experimental errors (ε). Similarly, the "Mean" row denotes the estimated coefficients of the model, capturing the "average" impact of each factor: Positive values indicate an increase in classification effectiveness and negative values indicate the opposite. Note that q_0 refers to the overall mean, computed using observations. The "99% CI" rows report the 99% confidence intervals associated with the overall mean q_0 and each effect q_f ($f \in \{A, B, AB\}$). Intervals that include zero indicate statistically nonsignificant impact of the associated factors. Finally, the "R^2" column reports the coefficient of determination of the proposed model: values close to 1 indicate a well-fitted model. Similar tables, referred to as ANOVA (analysis of variance) tables, are used to

TABLE 3. Factorial design applied to Rocchio, KNN, Naïve Bayes, and SVM for ACM-DL (*CD* × *TD* design: A = CD and B = TD. *CS* × *TD* design: A = CS and B = TD).

			Analysis of variance (ANOVA)					
	Model:		$y = q_0 + q_A x_A + q_B x_B + q_{AB}\, x_A x_B + \varepsilon$					
TC algorithm	Effects:		q_0	q_A	q_B	q_{AB}	ε	R^2
Rocchio	*CD*	%Var	–	69.98%	25.73%	0.15%	4.13%	0.96
	×	Mean	63.90	−10.41	−6.31	−0.48	–	
	TD	99% CI	[63.57, 64.23]	[−10.73, −10.08]	[−6.64, −5.98]	[−0.81, −0.16]	–	
	CS	%Var	–	69.52%	26.53%	0.08%	3.86%	0.96
	×	Mean	63.72	−10.27	−6.35	−0.36	–	
	TD	99% CI	[63.41, 64.04]	[−10.59, −9.96]	[−6.66, −6.03]	[−0.67, −0.04]	–	
KNN	*CD*	%Var	–	66.91%	28.71%	0.00%	4.38%	0.96
	×	Mean	63.02	−10.70	−7.01	0.04	–	
	TD	99% CI	[62.67, 63.38]	[−11.06, −10.34]	[−7.36, −6.65]	[−0.31, 0.40]	–	
	CS	%Var	–	65.31%	30.69%	0.03%	3.97%	0.96
	×	Mean	63.10	−10.50	−7.20	0.23	–	
	TD	99% CI	[62.76, 63.43]	[−10.84, −10.16]	[−7.53, −6.86]	[−0.11, 0.57]	–	
NB	*CD*	%Var	–	60.34%	35.17%	0.03%	4.45%	0.95
	×	Mean	62.04	−10.20	−7.79	−0.25	–	
	TD	99% CI	[61.68, 62.40]	[−10.56, −9.85]	[−8.15, −7.43]	[−0.61, 0.11]	–	
	CS	%Var	–	60.06%	35.59%	0.01%	4.34%	0.96
	×	Mean	62.03	−10.14	−7.80	−0.11	–	
	TD	99% CI	[61.68, 62.38]	[−10.49, −9.78]	[−8.16, −7.45]	[−0.46, 0.25]	–	
SVM	*CD*	%Var	–	65.87%	30.53%	0.63%	2.97%	0.97
	×	Mean	61.79	−12.59	−8.57	−1.23	–	
	TD	99% CI	[61.45, 62.14]	[−12.93, −12.24]	[−8.92, −8.22]	[−1.58, −0.88]	–	
	CS	%Var	–	64.07%	32.49%	0.49%	2.94%	0.97
	×	Mean	61.63	−12.26	−8.73	−1.07	–	
	TD	99% CI	[61.29, 61.97]	[−12.60, −11.92]	[−9.07, −8.39]	[−1.42, −0.73]	–	



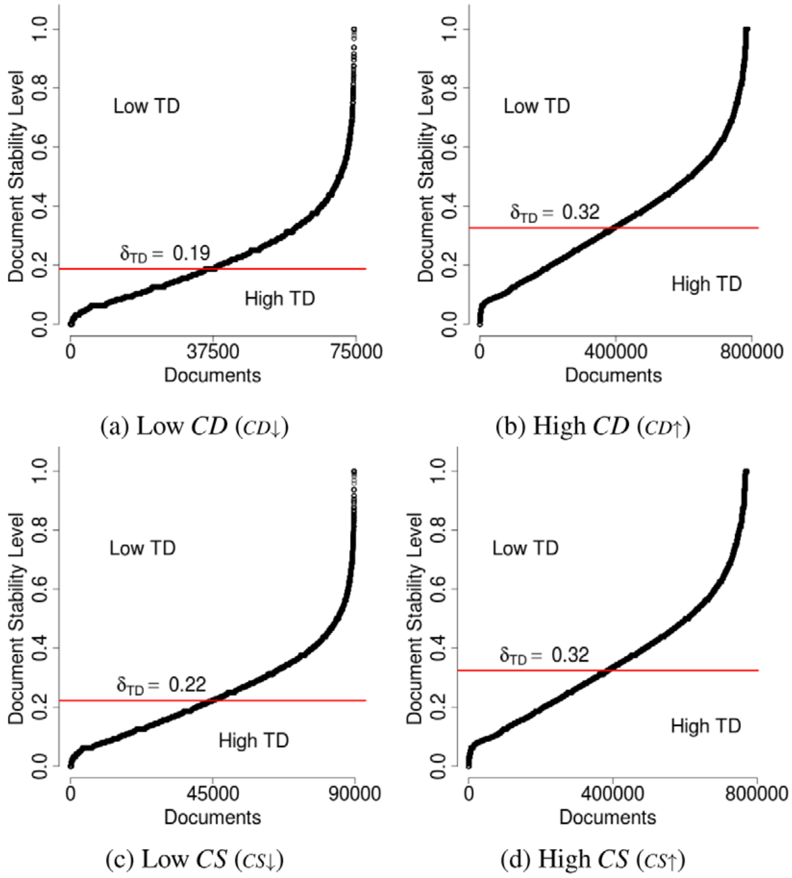
(a) Class Distribution Variation (*CD*) (b) Class Similarity Variation (*CS*)

FIG. 8. Determining the lower and upper levels of *CD* and *CS*—MEDLINE. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

summarize the results obtained with the other data sets as well. The final section provides a detailed discussion of all results.

MEDLINE. To build the four document partitions of each factorial design for the MEDLINE data set, we follow the same strategy adopted for ACM-DL. First, we partition the documents regarding the *CD* and *CS* factors, according to the δ_{CD} and δ_{CS} thresholds. These partitions and the corresponding thresholds are shown in Figure 8. We then further subdivide each of these partitions based on the *TD* factor, according to the δ_{TD} threshold, as depicted in Figure 9.

Note that, according to Figure 8, $|\mathbb{C}_{CD\uparrow} \cap \mathbb{C}_{CS\downarrow}| = 1$ and $|\mathbb{C}_{CD\downarrow} \cap \mathbb{C}_{CS\uparrow}| = 1$. Thus, the argument for the unfeasibility of a three-factor experimental design applied to ACM-DL



(a) Low *CD* ($CD\downarrow$)

(b) High *CD* ($CD\uparrow$)

(c) Low *CS* ($CS\downarrow$)

(d) High *CS* ($CS\uparrow$)

FIG. 9. Determining the lower and upper levels of *TD*—MEDLINE. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

also holds for MEDLINE. However, the figure also shows that three of the four classes with high *CS* also have high *CD*, and all classes with low *CS* also have low *CD*. Thus, once again, there is a high correlation between both factors, justifying our approach to decouple the three-factor design into two independent 2-factor analyses.

Because the MEDLINE data set is large (more than 800 thousand documents), we are able to replicate each experiment by performing a 10-fold cross-validation, as the test size is sufficiently large to achieve stable results among the replications. The results achieved with both factorial designs ($CD \times TD$ and $CS \times TD$), considering each TC algorithm, are summarized in Table 4 and analyzed in the Discussion section.

AG-NEWS. Finally, the same overall procedure is also adopted to build the two 2^2r experimental designs for AG-NEWS. We partition the documents with respect to the *CD* and *CS* factors using the δ_{CD} and δ_{CS} thresholds, as shown in Figure 10. We then further partition these groups according to the δ_{TD} threshold, as shown in Figure 11.

Similar to the other two data sets, Figure 10 shows that the number of documents in AG-NEWS is not enough to fill all partitions of the ideal 2^3 experimental design since $|\mathbb{C}_{CD\uparrow} \cap \mathbb{C}_{CS\downarrow}| = 1$ and $|\mathbb{C}_{CD\downarrow} \cap \mathbb{C}_{CS\uparrow}| = 1$. Thus, the lack of enough samples to fill the $CD\downarrow CS\uparrow$ partition prevents us from performing a complete three-factor design. The figure also shows that five of the six classes with high *CS* also have high *CD*, and four of the five classes with low *CS* also have low *CD*, indicating, once again, that both factors are correlated.

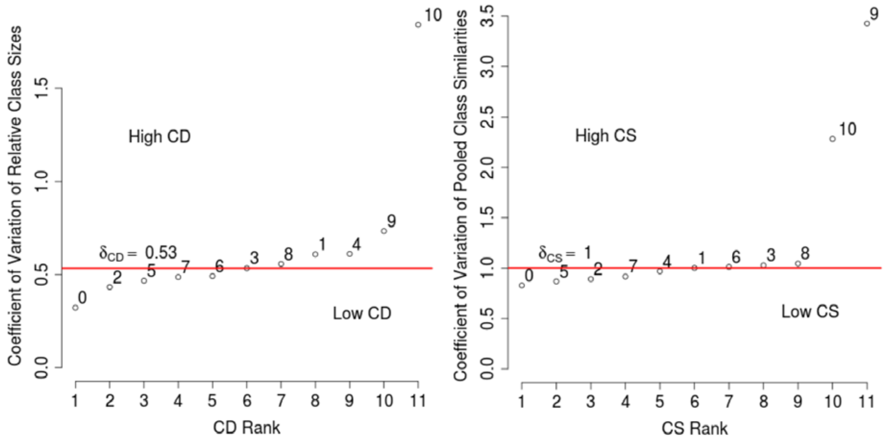
We replicate each experiment by performing a 10-fold cross-validation, because, similar to MEDLINE, AG-NEWS is also a large data set. Table 5 summarizes the results, which are discussed in the next section.

Discussion

Having presented our method to analyze the impact of temporal effects on TC algorithms and illustrated its applicability to four algorithms and three reference data sets, we now discuss our results, reported in Tables 3–5. Recall that, when analyzing the results of a specific experimental design, the impact of each factor on the response variable is captured by the percentage of variation explained by it ("% Var" in the ANOVA tables). However, when comparing results across different designs, as we do here, it is important also to analyze the effects associated with each factor *f*, q_f,

TABLE 4. Factorial design applied to Rocchio, KNN, Naïve Bayes, and SVM for MEDLINE ($CD \times TD$ design: A = CD and B = TD, $CS \times TD$ design: A = CS and B = TD).

			Analysis of variance (ANOVA)					
	Model:		$y = q_0 + q_A x_A + q_B x_B + q_{AB}\, x_A x_B + \varepsilon$					
TC algorithm	Effects:		q_0	q_A	q_B	q_{AB}	ε	R^2
Rocchio	*CD*	%Var	–	89.91%	9.02%	0.38%	0.69%	0.99
	×	Mean	79.74	–6.28	–1.99	0.41	–	
	TD	99% CI	[79.49, 79.99]	[–6.53, –6.03]	[–2.24, –1.74]	[0.16, 0.66]	–	
	CS	%Var	–	90.19%	7.93%	0.47%	1.41%	0.99
	×	Mean	81.38	–4.52	–1.34	0.32	–	
	TD	99% CI	[81.12, 81.63]	[–4.77, –4.26]	[–1.60, –1.08]	[0.07, 0.58]	–	
KNN	*CD*	%Var	–	69.60%	29.36%	0.13%	0.90%	0.99
	×	Mean	84.78	–3.34	–2.17	0.15	–	
	TD	99% CI	[84.61, 84.95]	[–3.52, –3.17]	[–2.34, –2.00]	[–0.03, 0.32]	–	
	CS	%Var	–	53.84%	43.24%	0.80%	2.12%	0.98
	×	Mean	86.52	–1.88	–1.68	0.23	–	
	TD	99% CI	[86.35, 86.69]	[–2.05, –1.71]	[–1.85, –1.52]	[0.06, 0.40]	–	
NB	*CD*	%Var	–	75.76%	23.27%	0.08%	0.88%	0.99
	×	Mean	86.18	–4.26	–2.36	–0.14	–	
	TD	99% CI	[85.98, 86.39]	[–4.47, –4.05]	[–2.57, –2.15]	[–0.35, 0.07]	–	
	CS	%Var	–	66.54%	31.74%	0.00%	1.72%	0.98
	×	Mean	87.56	–2.66	–1.84	–0.00	–	
	TD	99% CI	[87.37, 87.76]	[–2.85, –2.47]	[–2.03, –1.64]	[–0.20, 0.19]	–	
SVM	*CD*	%Var	–	69.22%	29.71%	0.05%	1.02%	0.99
	×	Mean	84.01	–4.09	–2.68	0.11	–	
	TD	99% CI	[83.78, 84.23]	[–4.32, –3.87]	[–2.91, –2.46]	[–0.11, 0.34]	–	
	CS	%Var	–	60.55%	37.17%	0.62%	1.66%	0.98
	×	Mean	86.35	–2.58	–2.02	0.26	–	
	TD	99% CI	[86.15, 86.54]	[–2.78, –2.39]	[–2.22, –1.83]	[0.07, 0.46]	–	



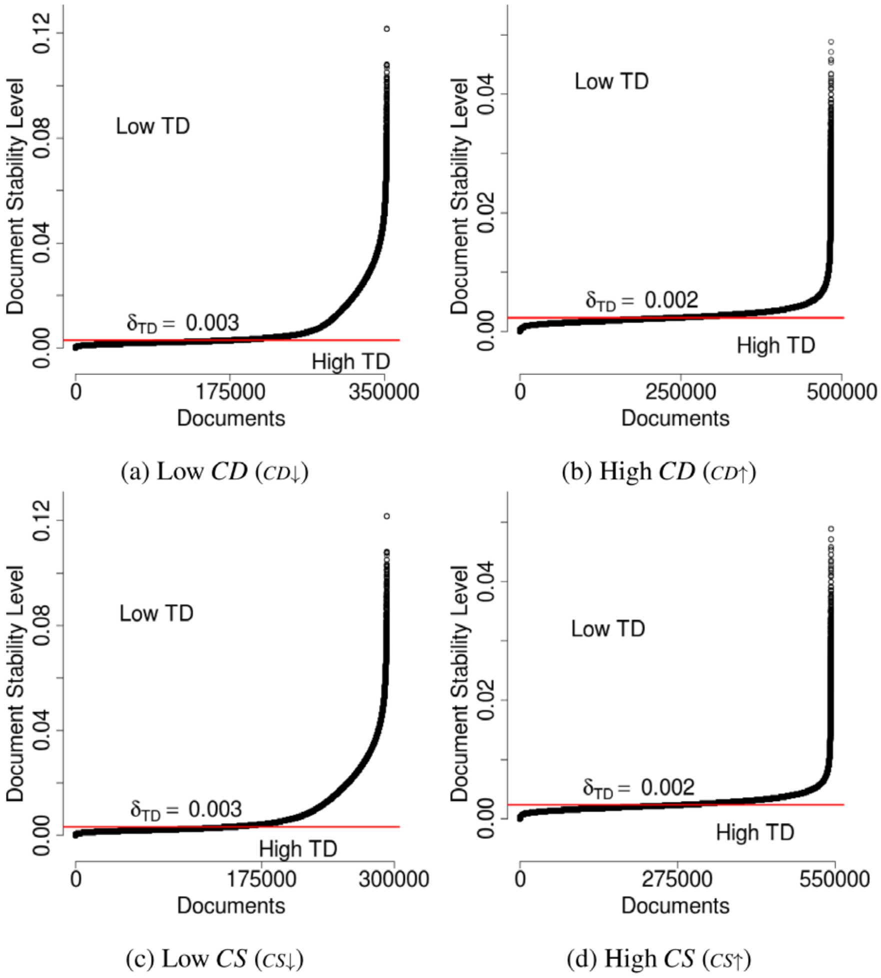
(a) Class Distribution Variation (*CD*)

(b) Class Similarity Variation (*CS*)

FIG. 10. Determining the lower and upper levels of *CD* and *CS*—AG-NEWS. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

and their relative impact on the overall mean q_0. Because the total variation of the responses (SS_T) may vary across different designs, the relative impact of each q_f on the overall mean q_0 allows a fairer comparison of the impact of each factor on the results across the designs. Ultimately, it represents the extent to which classification effectiveness improves or degrades, depending on the sign of q_f, when factor f is at its higher or lower level.

We start with two general observations. First, across all reference data sets and TC algorithms, our experimental designs are successful in isolating the parameters that are the target of the study: The analyzed temporal effects explain



(a) Low *CD* (CD↓)

(b) High *CD* (CD↑)

(c) Low *CS* (CS↓)

(d) High *CS* (CS↑)

FIG. 11. Determining the lower and upper levels of *TD*—AG-NEWS. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the vast majority of the variations observed in the results. Indeed, the percentages of variation left unexplained and thus credited to experimental errors (column ε) are under 5%, 2.2%, and 0.1% for ACM-DL, MEDLINE, and AG-NEWS, respectively. The larger variations left unexplained for the ACM-DL data set are possibly due to the fact that this data set is much smaller than the other two (small sample sizes result in greater variability). However, as we can observe in Table 3, the percentages of variation credited to experimental errors are inferior to the percentages credited to the temporal factors. Consistently, R^2 is greater than or equal to 0.95 in all cases.

Our second general observation is that the percentages of variation explained by the secondary factors (column q_{AB}), that is, the interactions between *CD* and *TD* in the $CD \times TD$ design, and between *CS* and *TD* in the $CS \times TD$ design, are very small across all data sets and algorithms, falling below 0.7%, 0.9%, and 2.7% for the ACM-DL, MEDLINE, and AG-NEWS data sets, respectively. Indeed, the effect of this interaction is statistically insignificant, with 99% confidence, in many of these cases (see line "99% CI"—the intervals including zero comprise statistically insignificant effects). If significant, the effect associated with the interaction is often negative, implying that it contributes to a degradation in the classification effectiveness, although the magnitude of such degradation is very small (up to 1.23% and 2.26%, on average, with respect to the overall average performance reported in q_0, for ACM-DL, AG-NEWS, respectively—in the MEDLINE data set, this was not observed). Only in a few cases is the effect of the interaction positive, implying that it actually contributes to improve classification effectiveness. We conjecture that this is a side effect of the interactions between *CD* and *CS* factors that are not captured by our pairwise experimental designs. In other words, the positive interaction is possibly due to a few classes having CD↓ and CS↑ in all three data sets, as argued above. Nevertheless, even when positive, the effect due to the interaction of multiple factors is very small, with an impact on the overall mean by as much as only 0.41% and 2.63%, on average, in MEDLINE and AG-NEWS, respectively—in the ACM-DL data set this was not observed. Thus, we argue that the primary factors *CD*, *CS*,

TABLE 5. Factorial design applied to Rocchio, KNN, Naïve Bayes, and SVM for AG-NEWS ($CD \times TD$ design: A = CD and B = TD, $CS \times TD$ design: A = CS and B = TD).

			Analysis of variance (ANOVA)					
	Model:		$y = q_0 + q_A x_A + q_B x_B + q_{AB}\, x_A x_B + \varepsilon$					
TC algorithm	Effects:		q_0	q_A	q_B	q_{AB}	ε	R^2
Rocchio	*CD*	%Var	–	49.78%	50.07%	0.11%	0.04%	0.99
	×	Mean	72.54	−10.14	−10.17	0.48	–	
	TD	99% CI	[72.42, 72.66]	[−10.26, −10.02]	[−10.29, −10.05]	[0.36, 0.61]	–	
	CS	%Var	–	53.40%	42.98%	3.57%	0.05%	0.99
	×	Mean	71.06	−10.16	−9.11	2.63	–	
	TD	99% CI	[70.91, 71.20]	[−10.30, −10.02]	[−9.26, −8.97]	[2.49, 2.78]	–	
KNN	*CD*	%Var	–	76.82%	20.95%	2.21%	0.02%	0.99
	×	Mean	80.16	−13.35	−6.97	−2.26	–	
	TD	99% CI	[80.05, 80.27]	[−13.45, −13.24]	[−7.08, −6.86]	[−2.37, −2.15]	–	
	CS	%Var	–	78.07%	21.48%	0.40%	0.04%	0.99
	×	Mean	80.11	−11.69	−6.13	−0.84	–	
	TD	99% CI	[79.99, 80.23]	[−11.81, −11.57]	[−6.25, −6.01]	[−0.96, −0.72]	–	
NB	*CD*	%Var	–	64.94%	34.98%	0.03%	0.04%	0.99
	×	Mean	79.15	−11.14	−8.18	0.25	–	
	TD	99% CI	[79.02, 79.28]	[−11.27, −11.02]	[−8.31, −8.05]	[0.13, 0.38]	–	
	CS	%Var	–	78.19%	21.11%	0.65%	0.06%	0.99
	×	Mean	79.30	−11.65	−6.05	1.06	–	
	TD	99% CI	[79.16, 79.44]	[−11.80, −11.51]	[−6.20, −5.91]	[0.92, 1.20]	–	
SVM	*CD*	%Var	–	81.38%	17.85%	0.74%	0.03%	0.99
	×	Mean	78.56	−14.76	−6.91	−1.41	–	
	TD	99% CI	[78.44, 78.69]	[−14.89, −14.63]	[−7.04, −6.79]	[−1.53, −1.28]	–	
	CS	%Var	–	77.36%	22.50%	0.07%	0.08%	0.99
	×	Mean	78.23	−12.43	−6.70	0.36	–	
	TD	99% CI	[78.05, 78.42]	[−12.61, −12.25]	[−6.89, −6.52]	[0.18, 0.54]	–	

and *TD* are the main sources of impact on classification effectiveness across all analyzed scenarios, focusing our discussion on them.

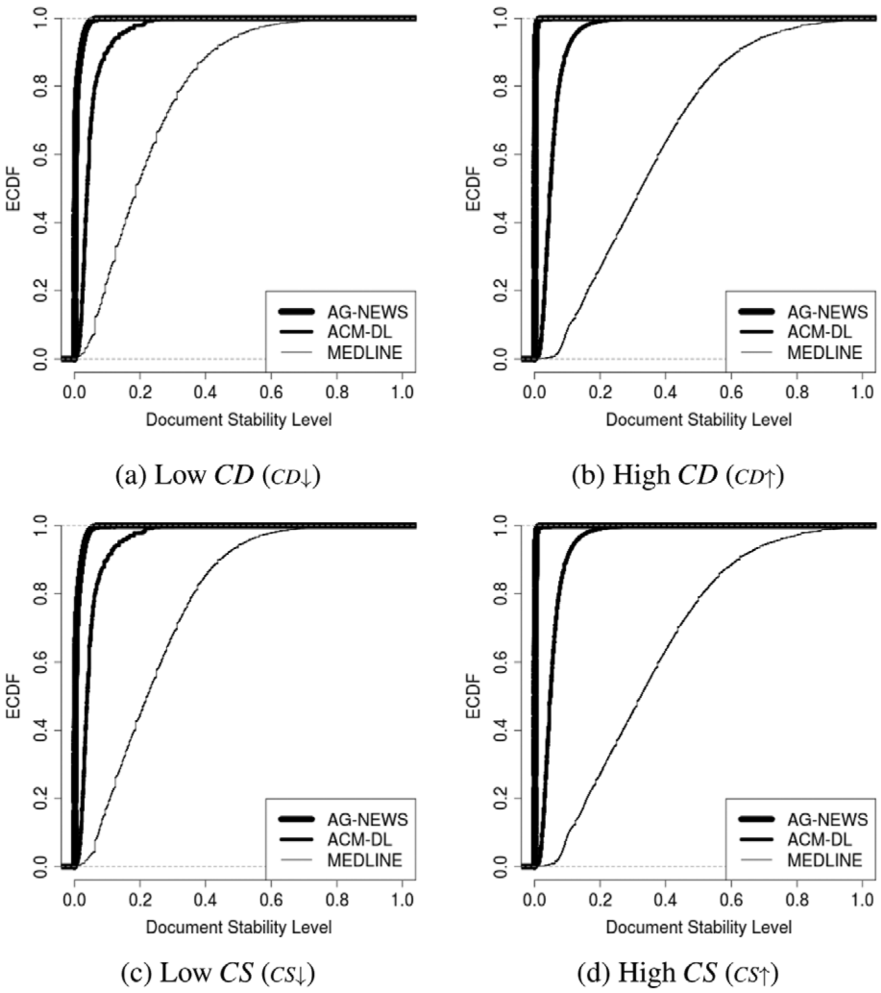
In the following, we analyze specific results for each reference data set, discussing the overall behavior observed across all TC algorithms in the next section. We then discuss the results for each specific TC algorithm, pointing out invariants across data sets and drawing insights from the influence of the temporal effects on each algorithm in the section following the next. Finally, we summarize the main implications of our findings.

Impact of Temporal Effects on the Reference Data Sets

We start by analyzing the relative impact of the temporal factors (q_f) on the average effectiveness q_0 of the TC algorithms in each data set, given by the ratio $\frac{q_f}{q_0}$ (also referred to as average performance degradation of the temporal factor *f*). As Tables 3–5 show, the effects associated with the temporal factors represent an impact on the average effectiveness of the TC algorithms that falls, on average, between 9.87% and 20.37% in the ACM-DL data set, 1.65% and 7.88% in MEDLINE, and 7.63% and 18.79% in AG-NEWS. Thus, the impact of the temporal effects on classification is much higher in the ACM-DL and AG-NEWS data sets than in the MEDLINE data set. Such observation is consistent with the characterization reported previously, which indicates a more stable behavior of MEDLINE in contrast to the more dynamic nature of ACM-DL and AG-NEWS. Considering the ACM-DL and MEDLINE data sets, it is also consistent with the qualitative analysis reported in Mourão et al. (2008), which showed that: (a) once a term appears, it tends to remain more stable over time in MEDLINE than in the other two data sets, thus implying a smaller impact of *TD* on classification of the former; and (b) the more consolidated knowledge area captured in the MEDLINE data set justifies the smaller impact of *CD* and *CS* on it.

To further corroborate such findings, we performed a two-sided Mann–Whitney test (Hollander & Wolfe, 1999) to compare the coefficient of variations (CVs) of class sizes (i.e., regarding *CD*) computed for the three reference data sets.[13] Recall that the CV values are reported in Figures 6a, 8a, and 10a for ACM-DL, MEDLINE, and AG-NEWS, respectively. With 99% confidence, we found that the CVs of class sizes in the MEDLINE data set are indeed smaller than those computed for the ACM-DL and AG-NEWS data sets (*p*-values of .001 and .005, respectively). Comparing the CV values computed for ACM-DL and AG-NEWS, we found that both samples are statistically indistinguishable, with that confidence level (*p*-value of .24). Thus, we state

[13] We chose a nonparametric test because the CV values regarding the CD aspect are not normally distributed.



(a) Low *CD* ($CD\downarrow$)

(b) High *CD* ($CD\uparrow$)

(c) Low *CS* ($CS\downarrow$)

(d) High *CS* ($CS\uparrow$)

FIG. 12. Cumulative distribution function of document stability level values. Curves near the upper left corner indicate the presence of more documents with lower DSL values (i.e., composed mostly of terms with varying distribution). Both ACM-DL and AG-NEWS data sets have a strong bias toward less stable terms, being more impacted by the *TD* effect. On the other hand, the less accented curve shows the more stable nature of the MEDLINE data set regarding its term distribution.

that $CD_{MEDLINE} < CD_{ACM–DL} \sim CD_{AG–NEWS}$ to refer to the relative impact of *CD* in each data set.

The same test was performed to compare the CVs of pooled similarities (i.e., related to *CS*), reported in Figures 6b, 8b, and 10b. Once again, we found that, with 99% confidence, the *CV* values computed for the MEDLINE data set are smaller than those obtained for ACM-DL and AG-NEWS (*p*-values of .005 and .0006, respectively), whereas no statistical difference was observed between the values computed for ACM-DL and AG-NEWS (*p*-value of .15). Thus, we state that $CD_{MEDLINE} < CD_{ACM–DL} \sim CD_{AG–NEWS}$.

To compare the impact of *TD* on the three data sets, we show, in Figure 12, the empirical cumulative distribution of the observed document stability level (DSL) values, for each level of *CD* and *CS* and for each reference data set. The curves for MEDLINE show a clear bias toward higher DSL levels, thus indicating a smaller impact of *TD*. The curves for both ACM-DL and AG-NEWS exhibit a much stronger bias towards less stable documents, exposing the more dynamic nature of these data sets. We note that, for the MEDLINE data set, the bias towards more stable documents is somewhat stronger for the $CD\uparrow$ and $CS\uparrow$ levels. In other words, the partitions with higher temporal variations in *CD* and *CS* tend to have more stable documents, in comparison with the partitions with lower variations on the two effects. This behavior is a peculiarity of the MEDLINE data set, being not observed in neither ACM-DL nor AG-NEWS data sets. Once again, we applied the Mann–Whitney test, finding that, with 99% confidence, the DSL values are indeed larger for the MEDLINE than for ACM-DL and AG-NEWS (*p*-values smaller than 10^{-5}), and that DSL values are larger in ACM-DL than in AG-NEWS (*p*-value $<10^{-5}$). Thus, we state that $TD_{MEDLINE} < TD_{ACM–DL} < TD_{AG–NEWS}$.

Having compared the relative impact of each temporal aspect across the three data sets, we now analyze the relative impact of the three temporal effects for each data set. A general observation is that, with 99% confidence, the relative impact of *TD* is consistently lower than the impact of

CD (or *CS*) on all four TC algorithms, considering the three explored textual data sets. In the case of ACM-DL, the impact of *TD* is up to 65% smaller than the impact of *CD*, and almost 62% times smaller than the impact of *CS*, considering the Rocchio classifier. In the case of MEDLINE and AG-NEWS, such a difference is more pronounced. Considering the MEDLINE data set, the impact of *TD* is more than 3 times smaller than the impact of *CD* and *CS* effects. In the case of AG-NEWS, such a difference almost doubles. One exception to this general behavior is observed when Rocchio is applied in the AG-NEWS data set: We cannot distinguish, with 99% confidence, the relative impact of *CD* from the relative impact of *TD*.

Note that, except for Rocchio on the AG-NEWS data set, *CD*'s impact on classification is higher than *TD*'s impact if and only if the *CS*'s impact is also higher than *TD*'s. This should come as no surprise, given the strong positive correlation between both factors, as discussed above. We reach these findings by analyzing the 99% confidence intervals for the effects of each factor ("99% CI" lines in Tables 3–5). Similar conclusions are reached by analyzing the percentages of variations explained by each individual factor (columns q_A and q_B of the mentioned tables). These findings reveal the challenges imposed by the temporal effects, and shows that developing strategies to handle them in TC algorithms is a promising research direction.

Impact of Temporal Effects on the TC Algorithms

We now turn our attention to the impact of the temporal effects on the TC algorithms. As we can observe from the three ANOVA tables, all three factors have negative effects (columns q_A and q_B) in all analyzed scenarios, implying that all explored TC algorithms are negatively impacted by the temporal effects in all three data sets. In fact, relative to the overall average performance (q_0), the effect of *CD* contributes to an average decrease in classification effectiveness by as much as 20.37% (for the SVM classifier). Similarly, higher levels of *CS* and *TD* contribute to a classification degradation of as much as 19.89% and 14.16% (also for the SVM classifier), on average. Moreover, the degradation is more significant for the reference data sets in which the impact of the temporal effects is stronger, that is, ACM-DL and AG-NEWS, as expected. Next, we discuss the impact on each specific algorithm, focusing on the results for the AG-NEWS data set (Table 5), as it is the one most influenced by all three temporal effects.

Starting with the Rocchio classifier, we observe that all three temporal effects greatly impact classification effectiveness, with more than 49% of the observed variations being explained by either the *CD* or the *CS* effects, and at least 43% (and up to 50%) being explained by the *TD* effect. Indeed, the factors contribute to a significant degradation in the overall classification effectiveness, in each design. For instance, in the $CD \times TD$ design a higher level of *CD* incurs an average degradation of 13.98% in the average performance, whereas the degradation caused by a higher level of *TD* is 14.02%, on average. Similarly, in the $CS \times TD$ design the degradation due to higher levels of *CS* and *TD* are, on average, 14.30% and 12.92%, respectively. The reasons for such a significant impact on Rocchio's performance are the following. *CD* and *CS* affect the coordinates of the centroids learned by the Rocchio classifier: as Miao and Kamel (2011) pointed out, the centroid vector does not take the distribution of class sizes into account, and thus may be affected by variations in such distribution. Because the distribution of class sizes in the entire data set may not be the same as the corresponding distribution observed when the test document was created, the classifier's prediction may be error prone. *TD* also significantly affects this classifier, because, when averaging the vectors of each class to compute the class centroids, it considers all training points to determine the class of a test document. Thus, it may be affected by the variations in the term-class relationships.

Similarly, both KNN and Naïve Bayes classifiers are also greatly impacted by the three temporal effects. The KNN classifier showed more than 76% of the observed variations allocated to the *CD*/*CS* effects, and more than 20% allocated to the *TD* effect. Similarly, considering the Naïve Bayes classifier, 65% (78%) of the observed variations were allocated to the *CD* (*CS*) effect, and at least 21% (and up to 35%) allocated to the *TD* effect. Indeed, both classifiers have a bias regarding the distribution of class sizes. In the KNN classifier, larger classes tend to have more documents in the *K*-neighbor set for each test document (Tan, 2005). The Naïve Bayes classifier, in turn, tends to privilege larger classes due to the class prior probability expressed in Equation 1: When the class conditionals are similar, this classifier uses a prior information to break ties, being directly affected by variations in the distribution of class sizes. Thus, similar to Rocchio, *CD* affects both classifiers' decision boundaries: since the distribution of class sizes considering the entire training set may not reflect the distribution when the test document was created, both classifiers may make wrong predictions. In fact, the *CD* effect incurs in an average performance degradation of KNN and Naïve Bayes of 16.65% and 14.07%, respectively. Moreover, *CS* affects the KNN classifier (with an average decrease of 14.59% in the average performance) because it directly perturbs the K nearest neighbor set, that is, because of differences in the pairwise class similarities this set may be composed of classes that were similar at different points in time. Naïve Bayes, in turn, is affected by *CS* (with an average decrease of 14.69% in the average performance) as this classifier considers a somewhat local metric to assess the relationships between terms and classes, expressed by the term conditional probability $P(t|c)$. As discussed above, estimating $P(t|c)$ ultimately searches a subset of the class vocabularies and, when vocabularies change over time the decision rule also changes. Finally, *TD* significantly impacts both classifiers because they consider the terms present in all training points to determine the decision boundaries. In KNN, the impact of *TD* takes place when building the *K*-neighbor set, whereas in Naïve Bayes such impact occurs when estimating the maximum

TABLE 6. A comparative study on the impact of the temporal effects on each TC algorithm—Rocchio (RO), SVM, Naïve Bayes (NB), and KNN. The algorithms are ordered from most affected by temporal effects for the least affected, for each collection and factor analyzed.

Temporal	Data set		
Effect	ACM-DL	MEDLINE	AG-NEWS
CD	SVM > KNN ~ NB ~ RO	RO > NB ~ SVM > KNN	SVM > KNN > NB ~ RO
CS	SVM > KNN ~ NB ~ RO	RO > NB ~ SVM > KNN	SVM > KNN ~ NB > RO
TD	SVM > NB > KNN > RO	SVM > NB > KNN > RO	RO > NB > SVM > KNN

likelihood estimates (MLE) from the training set—more specifically, the term conditional probabilities $P(t|c)$. Thus, both classifiers are also sensitive to variations in the term-class relationships.

Considering SVM, both *CD* and *CS* explain, each, more than 78% of the variations in classification effectiveness in both experimental designs. *TD*, in contrast, is responsible for at most 22% of the observed variation. The reasons for this behavior are the following. First, variations in the distribution of class sizes lead to boundary hyperplane skewness (see Sun, Lim, & Liu, 2009), potentially misleading the classification decisions when considering data distributed over several points in time with changing distribution. The solution to the SVM optimization problem must satisfy the so-called Karush-Kuhn-Tucker (KKT) conditions, which include the following (see Hastie, Tibshirani, & Friedman, 2009):

$$0 = \sum_{i=1}^{N} \alpha_i y_i \tag{4}$$

$$\beta = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i, \tag{5}$$

$$0 = \sum_{i=1}^{N} \alpha_i [y_i(x^T \beta + \beta_0) - 1]. \tag{6}$$

Due to the condition expressed by Equation 4, the increase of some α_i at the positive side of the hyperplane forces an increase of some α_i at the negative side to satisfy that constraint and, due to possible imbalances in the distribution of class sizes, either of them may receive a higher value, causing the hyperplane to be skewed towards the smaller class. Thus, clearly, *CD* does have a strong impact on this classifier, and so does *CS*, given that the two effects are strongly correlated. Regarding the *TD* effect, notice that the determination of β and β_0 requires the use of the entire training data, regardless of the creation times, and so does the definition of the support points[14] (see Equation 6), being impacted by the *TD* effect. Furthermore, during the classification phase, the support points are taken into account to determine the classes, again regardless of their creation times. As expressed by Equation 5, the support points, with $\alpha_i > 0$, affect the decision rule $\hat{F} = sign(\vec{x}^T \vec{\beta} + \beta_0)$, thus ultimately being susceptible to the impact of the *TD* effect.

Turning our attention to the results for MEDLINE and AG-NEWS, reported in Tables 4 and 5, we find that, as observed with ACM-DL, the impact of *CD* and *CS* on all four TC algorithms are consistently higher than the impact of *TD*. This should come as no surprise because, as discussed in the Impacts of the Temporal Effects of the Reference Data Sets section, these data sets are more influenced by the *CD*/*CS* effects than by *TD*. Moreover, the discussion just presented, regarding each of the TC algorithms, also applies here and so will not be repeated.

We summarize our findings on the impact of the temporal effects on the four TC algorithms in Table 6, which shows a partial ordering of the algorithms with respect to the average impact of each temporal effect for each data set. This ordering is determined by taking the overall average performance of each algorithm (q_0) as baseline, analyzing the effect associated with each factor q_f (along with its corresponding 99% confidence interval), and its relative difference to the overall average. We stress here that such partial ordering is not linked to the effectiveness of each classifier, but with their sensibility to the temporal effects. As we can see, the SVM classifier is the most affected by the *CD* and *CS* effects in the (more dynamic) ACM-DL and AG-NEWS data sets, and the most affected by the *TD* effect in the MEDLINE data set. On the other hand, the Rocchio classifier was the most affected by *CD* and *CS* effects in the MEDLINE data set, being the most affected by the *TD* effect in the AG-NEWS. The other two classifiers (Naïve Bayes and KNN) were also affected, but to a smaller extent, compared to Rocchio and SVM. These relationships reinforce that, apart from being negatively impacted by all three temporal effects, the four explored TC algorithms exhibit distinct behavior when faced with data sets with specific temporal dynamics, as revealed by the conducted factorial designs. We believe that such analysis reinforces the challenges regarding the temporal effects issue, which depends not only on the data set at hand but also on the TC algorithm employed.

Implications

The analyses performed in the previous sections provide some general guidelines regarding the definition of requirements for strategies to deal with temporal effects in TC.

[14] The training instances $\vec{x}_i$ with corresponding $\alpha_i > 0$ lie in the class boundaries and are the so-called support points.

First, these strategies should consider stable terms, since they untangle some latent structural properties of the classes. It may be tempting to consider just training documents created at (or nearby) the creation time of the test document (window-based approach) to define class boundaries. However, such a strategy may not be a wise choice because it may lead to data sparseness problems. Moreover, it may also discard valuable information regarding stable terms occurring in training documents created at points in time other than the test document's creation time—which may reveal discriminative evidence about the classes' structural properties. Even when considering training documents created at different moments with respect to the test document, stable terms may still provide valuable information to the classifier. Such information, however, is neglected when adopting window-based strategies. This justifies the use of instance weighting strategies, especially when dealing with more stable data sets, such as MEDLINE. However, in order for this strategy to be successful the weighting function must capture the underlying process that guides the temporal evolution of the data set. Furthermore, not only the stability of the terms over time should be explored but also the variations in the distributions of class sizes and class similarities.

Similarly, the proposed method to evaluate the impact of the temporal effects on classification effectiveness provides valuable insights to better understand the behavior not only of the considered TC algorithms when faced with these effects but also of strategies aimed at overcoming them. We now turn our attention to this latter aspect.

We delineate below a general strategy for handling the temporal effects in TC. We start by considering the term distribution effect by means of a general instance weighting schema which directly deals with such effect by reweighting training document x according to the temporal distance between its creation time and that of a test document x'. Clearly, because these weights depend on the creation time of the test document, a lazy learning strategy would be necessary, as depicted in Algorithm 2. In the following algorithms, the function CreationTime(x) returns the creation time point p of x. Also, the set $\mathbb{D}_{trn}$ denotes the training set.

Algorithm 2. General instance weighting temporally aware classifier.

```
1. function $\textsc{Classify}(x', \mathbb{D}_{trn})$
2. $\mathbb{D}^{w}_{trn} \leftarrow \varnothing$
3. for all $x \in \mathbb{D}_{trn}$ do
4.     $x^{w} \leftarrow \delta(\textsc{CreationTime}(x), \textsc{CreationTime}(x')) \cdot x$
5.     $\mathbb{D}^{w}_{trn} \leftarrow \mathbb{D}_{trn} \cup \{x^{w}\}$
6.  $h \leftarrow \textsc{Train}(\mathbb{D}^{w}_{trn})$
7. return $h(x')$
```

In Algorithm 2, each training document x is reweighted according to a function $\delta : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ (see line 4), which we call *temporal weighting function*. Such a function captures the variation in the underlying data distribution observed over the timepoints spanning from CreationTime(x) and CreationTime(x'). The reweighted training documents are then used to learn a traditional classifier h, which is used to classify x'. As our previous analyses indicate, δ is data set-specific and must be carefully designed.

To also consider the class distribution and class similarity effects, we delineate a second general strategy which explores the fact that the temporal effects can be safely neglected when learning a classifier with data from a single point in time, at the cost of more data sparseness and possible loss of discriminative power. To mitigate such drawbacks, we learn specific classifiers for each timepoint p independently and, when classifying x', we consider all learned classifiers, properly adjusted to minimize the impact of the observed data variations between p and CreationTime(x'). This general idea can be found in Algorithm 3.

Algorithm 3. General ensemble based temporally aware classifier.

```
function $\textsc{Classify}(x', \mathbb{D}_{trn})$
for all time point $p$ do
    $\mathbb{D}^{p}_{trn} \leftarrow \{x \in \mathbb{D}_{trn} \mid \textsc{CreationTime}(x) = p\}$
    $h^{p} \leftarrow \textsc{Train}(\mathbb{D}^{p}_{trn})$
return $\sum_{p} h^{p}(x')\delta(p, \textsc{CreationTime}(x'))$
```

A preliminary instantiation of these ideas is described in Salles et al. (2010), which proposed temporally aware algorithms that are more robust to the temporal effects, and applied them to both the ACM-DL and MEDLINE data sets. Such algorithms can be thought of as a first attempt to instantiate the ideas delineated in both general algorithms. In any case, the analyses and investigations performed here allow us now to have a deeper understanding of the results achieved in that work.

To further clarify such discoveries, we start by briefly describing two strategies, proposed in Salles et al. (2010), to derive the temporally aware classifiers. The first one, called temporal weighting on documents, weights each training document by a temporal weighting function (TWF) according to its temporal distance to the test document. The TWF is modeled according to the observed variations over time in the term-class relationships for each data set, ultimately addressing the *TD* aspect. The second strategy, called temporal weighting on scores, groups training documents into partitions composed of documents created at the same point in time, performing a traditional classification procedure over each partition. This second strategy assumes that, when considering unitary periods, the temporal effects may be safely neglected, the classification models learned for each point in time are not affected by them. However, as previously stated, considering only the data related to a single point in time may disregard valuable information to learn an accurate classification model. Thus, the second step of this strategy consists of aggregating the information learned for each point in time, weighting the obtained classification scores by the TWF. Aggregating the obtained scores for each point in time is affected by the *TD* aspect because the scores reflect the relationships between terms and classes. To overcome the observed variations in the term-class relationships across the different points in time, the TWF is used to weight them according to the temporal distance between the points in time associated with each partition and the creation time of the test document. Thus, although the first step addresses the *CD* and *CS* effects, the second step addresses the *TD* aspect observed when aggregating the scores.

Analyzing the results reported in that work, we can observe that, for ACM-DL, both strategies achieved significant gains. Considering the temporal weighting on documents approach, we can justify its gains due to the high impact of *TD* in that data set. Moreover, because ACM-DL is also subject to a high impact of both *CD* and *CS*, the temporal weighting on scores approach also performed well, since it addresses such effects when considering one moment at a time. In MEDLINE, in contrast, because the impact of *TD* is smaller than the impact of the other two effects, we should expect less significant gains achieved by temporal weighting on documents. Indeed, this was the observed behavior: such an approach achieved statistical ties compared to baselines in almost all cases. However, as both *CD* and *CS* pose as important factors for that data set, we can observe statistically significant improvements in classification effectiveness when temporal weighting on scores is applied. Furthermore, the largest improvements achieved with the TWF applied on documents were observed with the Naïve Bayes classifier which, according to the analysis, is the classifier most affected by the *TD* effect. The Rocchio and KNN classifiers enjoy even larger improvements when the temporal weighting on scores is applied, since both are also significantly affected by both the *CD* and *CS* effects.

Conclusions and Future Work

In this work, we proposed a method, based on a series of full factorial designs, to evaluate the impact of temporal effects on text classification algorithms when applied to several reference data sets. First, we extended the characterization performed by Mourão et al. (2008), providing evidence of the existence of three temporal effects in three textual data sets, namely ACM-DL, MEDLINE, and AG-NEWS. Then we instantiated the method to quantify the impact of the temporal effects on the classification effectiveness of four well-known TC algorithms: Rocchio, KNN, Naïve Bayes, and SVM.

Our characterization results show that, contrary to the assumption of the static data distribution on which all four explored algorithms are based, each reference data set has a specific temporal behavior, exhibiting changes in the underlying data distribution across time. Such temporal variations potentially limit the classification performance. According to our results, the ACM-DL and AG-NEWS data sets are much more dynamic than the MEDLINE data set, resulting in the four explored TC algorithms being more impacted by the temporal aspects in the first two data sets. In addition to such findings, our approach enabled us to quantify the impact of each temporal aspect on the analyzed data sets and algorithms, allowing us to answer the two following questions, posed in the Introduction:

1. *Which temporal effects are the strongest in each data set?* In the three explored data sets, the impact of the observed temporal variations in the distribution of class sizes and in the pairwise class similarities are statistically superior to the impact of the observed variations in the term distribution on all classifiers (Rocchio being an exception in the case of the AG-NEWS data set). The MEDLINE data set enjoys a more skewed distribution regarding the impact of these effects, being much more impacted by the first two temporal aspects than by the term distribution variation. These findings reveal the challenges imposed by the temporal effects and that developing strategies to handle them in TC algorithms is a promising research direction.
2. *What is the behavior of each TC algorithm when dealing with different levels of each temporal aspect?* All four explored TC algorithms suffer a negative impact of the temporal aspects in terms of classification effectiveness, being the most significant impacts observed when these algorithms are applied to the most dynamic data sets (i.e., ACM-DL and AG-NEWS). Both the Rocchio and SVM classifiers were shown to be less robust to these effects when compared to Naïve Bayes and KNN classifiers. Thus, the temporal dimension turns out to be an important aspect that has to be considered when learning accurate classification models.

The method can be improved in at least two ways. First, we consider it important to devise a strategy to capture the interactions between the *CD* and *CS* aspects, overcoming the limitations observed in the decoupled factorial designs. This is particularly interesting if we are dealing with data sets that do not have the high correlation between both aspects, as observed in the three reference data sets analyzed here. Second, one could define more fine-grained levels for the temporal aspects, in order to achieve an even more accurate quantitative analysis regarding their impact on the TC algorithms. We argue that, despite the higher computational demands, such fine-grained experimental designs allow us to gather a deeper understanding of the behavior of TC algorithms with regard to the temporal dynamics of the data, which could drive the design of more accurate classification models.

In a classification framework, not only the learning step may be affected by the temporal dynamics of data but also some of the data preprocessing steps, such as feature selection, document representation, and data sampling. For example, since some inductive TC algorithms are affected by the class imbalance problem, where some classes are more representative than others, it is a common strategy to preprocess the data to provide more balanced training sets. The usual way to balance the class distribution on training data consists of oversampling the smaller classes or undersampling the larger ones. However, to the best of our knowledge, none of the already proposed strategies for data balancing handle the temporal dimension. Thus, we plan to further study the impact of the temporal dynamics on class balancing strategies. This may reveal effective approaches to further improve such data processing strategies and, ultimately, lead to more accurate classification models. Another aspect that can be studied is the document representation by exploiting more complex features, not only bags of words or n-grams, but patterns of frequently co-occurring terms that are not necessarily contiguous and the evolution of these patterns over time.

Another very direct line of investigation for future work is to produce better instantiations of the general strategies delineated in Algorithms 2 and 3 and evaluate such instantiations as more robust solutions for the impact of the temporal effects in TC. Finally, one important direction for future work is how to generalize some of the posed research questions in order to cover broader data sets and algorithms and to provide further insights in this difficult problem of tackling the temporal effects and evolution of text collections in TC.

Acknowledgments

This research was partially funded by the Brazilian National Institute of Science and Technology for Web Research (MCT/CNPq/INCT Web Grant Number 573871/2008-6), and by the authors' individual grants from CNPq, CAPES, FINEP, and FAPEMIG.

References

Bifet, A., & Gavaldà, R. (2006). Kalman filters and adaptive windows for learning in data streams. In L. Todorovski, N. Lavrač, & K.P. Jantke (Eds.), Discovery science (pp. 29–40). Berlin Heidelberg: Springer.

Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In C. Apte, D. Skillicorn, B. Liu, & S. Parthasarathy (Eds.), Proceedings of the SIAM International Conference on Data Mining (pp. 443–448). SIAM.

Box, G.E.P., Hunter, W.G., & Hunter, J.S. (1978). Statistics for experimenters: An introduction to design, data analysis and model building. New York: John Wiley & Sons.

Caldwell, N.H.M., Clarkson, P.J., Rodgers, P.A., & Huxor, A.P. (2000). Web-based knowledge management for distributed design. IEEE Intelligent Systems, 15(3), 40–47.

Cohen, W.W., & Singer, Y. (1999). Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems, 17(2), 141–173.

de Lima, E.B., Pappa, G.L., de Almeida, J.M., Gonçalves, M.A., & Meira, Jr., W. (2010). Tuning genetic programming parameters with factorial designs. In Proceedings of the IEEE Congress on Evolutionary Computation (pp. 1–8). IEEE.

Dries, A., & Rückert, U. (2009). Adaptive concept drift detection. Statistical Analysis and Data Mining, 2(5–6), 311–327.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research, 9, 1871–1874.

Figueiredo, F., da Rocha, L.C., Couto, T., Salles, T., Gonçalves, M.A., & Meira, Jr., W. (2011). Word co-occurrence features for text classification. Information Systems, 36(5), 843–858.

Folino, G., Pizzuti, C., & Spezzano, G. (2007). An adaptive distributed ensemble approach to mine concept-drifting data streams. In Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (pp. 183–188). IEEE.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research, 3, 1289–1305.

Forman, G. (2006). Tackling concept drift by temporal inductive transfer. In Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval (pp. 252–259). ACM.

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In A.L.C. Bazzan & S. Labidi (Eds.), Proceedings of the Brazilian Symposium on Artificial Intelligence (pp. 286–295). Berlin Heidelberg: Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.), Springer Series in Statistics. New York: Springer.

Hollander, M., & Wolfe, D.A. (1999). Nonparametric statistical methods. New York: Wiley-Interscience.

Jain, R. (1991). The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling. New York: John Wiley & Sons.

Joachims, T. (1998). Text categorization with suport vector machines: Learning with many relevant features. In C. Nedellec & C. Rouveirol (Eds.), Proceedings of the European Conference on Machine Learning (pp. 137–142). London, UK: Springer-Verlag.

Joachims, T. (1999). Making large-scale support vector machine learning practical. In B. Schölkopf, C.J.C. Burges, & A.J. Smola (Eds.), Advances in kernel methods (pp. 169–184). Cambridge MA, USA: MIT Press Cambridge.

Kim, Y.S., Park, S.S., Deards, E., & Kang, B.H. (2004). Adaptive web document classification with MCRDR. In Proceedings of the International Conference on Information Technology: Coding and Computing (pp. 476–480). IEEE.

Klinkenberg, R. (2004). Learning drifting concepts: Example selection vs. example weighting. Intelligent Data Analysis, 8(3), 281–300.

Klinkenberg, R., & Joachims, T. (2000). Detecting concept drift with support vector machines. In P. Langley (Ed.), Proceedings of the International Conference on Machine Learning (pp. 487–494). Stanford, CA: Morgan Kaufmann Publishers Inc.

Klinkenberg, R., & Rüping, S. (2003). Concept drift and the importance of example. In J. Franke, G. Nakhaeizadeh, & I. Renz (Eds.), Text mining: Theoretical aspects and applications (pp. 55–78). New York: Physica-Verlag.

Kolter, J.Z., & Maloof, M.A. (2003). Dynamic weighted majority: A new ensemble method for tracking concept drift. Technical report, Department of Computer Science, Georgetown University, Washington, DC.

Kong, X., & Yu, P.S. (2011). An ensemble-based approach to fast classification of multi-label data streams. In Seventh International Conference on Collaborative Computing: Networking, Applications and Worksharing (pp. 95–104). IEEE.

Koychev, I. (2000). Gradual forgetting for adaptation to concept drift. In Proceedings of the ECAI Workshop Current Issues in Spatio-Temporal Reasoning (pp. 101–106). IEEE.

Kuncheva, L.I., & Žliobaite, I. (2009). On the window size for classification in changing environments. Intelligent Data Analysis, 13(6), 861–872.

Lawrence, S., & Giles, C.L. (1998). Context and page analysis for improved web search. IEEE Internet Computing, 2(4), 38–46.

Lazarescu, M.M., Venkatesh, S., & Bui, H.H. (2004). Using multiple windows to track concept drift. Intelligent Data Analysis, 8(1), 29–59.

Liu, A., Ghosh, J., & Martin, C. (2007). Generative oversampling for mining imbalanced data sets. In R. Stahlbock, S.F. Crone, & S. Lessmann (Eds.), Proceedings of the International Conference on Data Mining (pp. 66–72). Las Vegas, NV: CSREA Press.

Lin, Z., Hao, Z., Yang, X., & Liu, X. (2009). Several SVM ensemble methods integrated with under-sampling for imbalanced data learning. Lecture Notes in Computer Science, 5678. Advanced data mining and applications (pp. 536–544). New York: Springer.

Liu, R.-L., & Lu, Y.-L. (2002). Incremental context mining for adaptive document classification. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 599–604). New York: ACM.

Manning, C.D., Raghavan, P., & Schtze, H. (2008). Introduction to information retrieval. New York: Cambridge University Press.

DOI: 10.1002/asi

Miao, Y.-Q., & Kamel, M. (2011). Pairwise optimized Rocchio algorithm for text categorization. Pattern Recognition Letters, 32(2), 375–382.

Mourão, F., Rocha, L., Araújo, R., Couto, T., Gonçalves, M., & Meira, Jr., W. (2008). Understanding temporal aspects in document classification. In Proceedings of the International Conference on Web Search and Web Data Mining (pp. 159–170). New York: ACM.

Nigam, K., & McCallum, A. (1998). A comparison of event models for naïve bayes text classification. In Proceedings of the AAAI/ICML-98 Workshop on Learning for Text Categorization (pp. 41–48). AAAI Press.

Nishida, K., & Yamauchi, K. (2007). Detecting concept drift using statistical testing. In V. Corruble, M. Takeda, & E. Suzuki (Eds.), Proceedings of the International Conference on Discovery Science (pp. 264–269). Berlin Heidelberg: Springer.

Nishida, K., & Yamauchi, K. (2009). Learning, detecting, understanding, and predicting concept changes. In Proceedings of the International Joint Conference on Neural Networks (pp. 283–290). IEEE.

Orair, G.H., Teixeira, C., Wang, Y., Meira, Jr., W., & Parthasarathy, S. (2010). Distance-based outlier detection: Consolidation and renewed bearing. Proceedings of the VLDB Endowment, 3(2), 1469–1480.

Rocha, L., Mourão, F., Pereira, A., Gonçalves, M.A., & Meira, Jr., W. (2008). Exploiting temporal contexts in text classification. In Proceedings of the International Conference on Information and Knowledge Engineering (pp. 243–252). New York: ACM.

Salles, T., Rocha, L., Pappa, G.L., Mourão, F., Gonçalves, M.A., & Meira, Jr., W. (2010). Temporally-aware algorithms for document classification. In Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval (pp. 307–314). New York: ACM.

Scholz, M., & Klinkenberg, R. (2007). Boosting classifiers for drifting concepts. Intelligent Data Analysis, 11(1), 3–28.

Shuttleworth, M. (2014). Factorial design, research design. Retrieved from https://explorable.com/factorial-design

Sun, A., Lim, E.-P., & Liu, Y. (2009). On strategies for imbalanced text classification using svm: A comparative study. Decision Support Systems, 48(1), 191–201.

Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Systems with Applications, 28(4), 667–671.

Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College, Dublin, Ireland.

Vaz de Melo, P.O., da Cunha, F.D., Almeida, J.M., Loureiro, A.A., & Mini, R.A. (2008). The problem of cooperation among different wireless sensor networks. In Proceedings of the International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 86–91). New York: ACM.

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. Machine Learning, 23(1), 69–101.

Yang, C., & Zhou, J. (2008). Non-stationary data sequence classification using online class priors estimation. Pattern Recognition, 41(8), 2656–2664.

Yang, Y. (1994). Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 13–22). New York: ACM.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval (pp. 42–49). New York: ACM.

Zhang, Z., & Zhou, J. (2010). Transfer estimation of evolving class priors in data stream classification. Pattern Recognition, 43(9), 3151–3161.

Žliobaite, I. (2009). Combining time and space similarity for small size learning under concept drift. In J. Rauch, Z. Raś, & P. Berka (Eds.), Proceedings of the International Symposium on Foundations of Intelligent Systems (pp. 412–421). Berlin Heidelberg: Springer.