Change detection in textual classification with unexpected dynamics

Clément Christophe, Julien Velcin, Jairo Cugliari, Philippe Suignard, Manel Boumghar

Change detection in textual classification with unexpected dynamics

Clément Christophe[1,2], Julien Velcin[1], Jairo Cugliari[1], Philippe Suignard[2], and Manel Boumghar[2]

[1] Université de Lyon, Lyon 2, ERIC UR3083, France {julien.velcin, jairo.cugliari}@univ-lyon2.fr
[2] EDF R&D, Boulevard Gaspard Monge, 91120 Palaiseau, France
{clement.christophe,philippe.suignard,manel.boumghar}@edf.fr

Abstract. Identifying changes in the dynamics of a classification scheme is an important task to solve using textual data streams. Changes in the volume of documents classified into one category could be a sign of a new emerging structure, which therefore gives clues on the need to update the classification scheme. In this paper, we present a method based on forecasting techniques, change detection and time series monitoring in order to raise alerts as soon as a change occurs in the volume of a given category. We build features only based on the textual content that enable us to accurately predict the expected temporal evolution of such category. Then, we use statistical process control to determine if the current volume is too far away from the one we might expect. We test our method on the New York Times Annotated Corpus and on an industrial data set from Electricité de France (EDF) and we observe that it raises alerts at the right time compared to other techniques from the literature.

Keywords: Change Detection · Text Streams · Forecasting.

1 Introduction

Nowadays, companies all over the world receive a considerable amount of feedback from clients. A major part of these feedback is done by email and companies that receive large quantities of emails have implemented classification algorithms based on pre-defined classification schemes. These schemes are implemented by domain experts who know which type of feedback clients will send to the company. In other words, they expect some topics of discussion to be more present in the data, because they have a profound understanding of the field and of the market. However, these classification schemes may have to evolve in the future. Observing changes in a classification scheme for streaming data is a crucial task for companies to solve. An unexpected change in the volume of documents covered by one category or another can be a sign of a new emerging problem. Therefore it could mean that the classification scheme should be updated. On one hand, some categories may have a temporal pattern: they occur more frequently at the beginning of the year, month, or week, or during specific seasons. On the other hand, it is essential for companies to be able to detect as soon as possible if the frequency of a category is evolving unexpectedly over time.

While analyzing textual data streams, it is common to model the information in order to detect events, outliers or new topics (Pimentel et al., 2014). For example, methods like Allan et al., 2000; Long et al., 2011; Metzler, Cai, and Hovy, 2012; Huang, Peng, and Wang, 2015; Peng et al., 2018 are designed to raise an alert as soon as a different document is detected but the core contribution resides in the mathematical definition of this difference. This is why these methods are well suited for different tasks like event, outlier, first-story detection.

Authors of Online-Latent Dirichlet Allocation (OLDA) (Lau, Collier, and Baldwin, 2012) developed an unsupervised topic model that can be used to raise an alert as soon as the meaning of a latent topic is changed by the publication of documents. Finally, TopicSketch (Xie et al., 2016) is designed to raise an alert if the frequency of single words or their co-occurrences evolve in an unexpected manner. Even if these methods monitor different entities of the textual content, we notice that the transformation from text to time series is one of their main method in common. In this work, we compare our model to two baselines. The first is Allan et al., 2000 which has, according to their conclusion, hit the limit of simple *Information Retrieval* approaches. Partly because they are using only one feature (TF-IDF) to represent the textual content in a bag-of-words setting. The second is Xie et al., 2016 which is a state of the art method for the task of event detection using only textual content. These methods are built to detect sudden and short burst in the data and not long-term change.

The field of novelty detection in textual data is a rather recent field with an ill-posed problematic. While it has been studied in several manners and contexts (Markou and Singh, 2003; Tsai et al., 2011; Christophe et al., 2019), there is no existence of a general framework or evaluation methods. When looking at a textual data stream, we can consider different types of novelty: a neologism, a new topic, a change in the distribution of the data, etc. We separate the field of novelty detection into two distinct tasks:

- Volume Novelty: it is observed in known entities of the textual contents as words or topics. It resides in the temporal aspect of these entities. For example: the frequency of a word or of a known category can vary in unusual ways.
- Structure Novelty: it is observed when an underlying change in the distribution of the data is detected, or when it concerns unknown or abstract entities of the textual content. For example, with the appearance of a new word or of a new topic. Also, some complex changes can be observed: topics can merge together, the underlying distribution of a topic can be so modified that there is a change in the meaning.

While the two tasks are different in nature, they are often intimately linked: the change in the frequency of a certain word can be a sign of the appearance of a new topic and vice-versa. However, by considering these tasks distinctively, we can develop a framework to resolve them. In this work, we will focus on the resolution of the first type of novelty: Volume novelty. This type of novelty is easier to analyze in a quantitative point of view and is very present in our industrial use-case. We will work with two textual datasets: the New York Times Annotated Corpus (NYTAC) and the Électricité de France (EDF) dataset for industrial use. These datasets contain news articles written by the New York Times and anonymized emails sent by clients to EDF. Each document

is classified into a particular category of a classification scheme: manually for the New York Times and automatically for EDF. Volume of documents classified into each category at each time-step can vary over time so we will analyze this dynamic and detect unexpected changes.

In this work, we challenge a classification scheme already used by EDF. We do not want to detect abnormal values as outliers or anomalies but we expect to find an aggregation of these odd values that gives us information about the change of a category. This task is often solved in other fields of application: in industrial processes (El-Shal and Morris, 2000), healthcare (Noyez, 2009), product quality (MacGregor and Kourti, 1995), power monitoring (Lazzaretti et al., 2016) or cybersecurity (Tartakovsky, Polunchenko, and Sokolov, 2012). A popular choice among these methods is the use of sequential analysis algorithm like the Cumulative Sum Control Chart (CUSUM) (Page, 1954) to raise an alert when the signal differs from previous (labelled as "normal") behavior. These methods use time series monitoring to raise alerts when changes occur. They are not well-suited for textual data and they do not necessarily use forecasting algorithms. We aim to solve our problem in two parts, first by forecasting the evolution of the monitored signal and then by analyzing the forecasting error. Our main objectives are as follows:

1. to **learn the underlying dynamics** of categories inside a classification scheme.
2. to **raise an alert** when a learned dynamic seems to change.
3. to be **as quick as possible** to detect the change.
4. to **limit the number of false alarms** in term of extreme value: we do not want to raise alarms for one-time changes.

In this paper, we present an algorithm able to raise alerts as soon as a non-predicted change in the volume of documents labelled into one category is detected. We assume that the documents are automatically labelled into pre-defined categories by a classification scheme. In Section 2, we start by detailing how we use exogenous variables to forecast a signal and how we apply the CUSUM algorithm to raise alerts. In Section 3, we present our dataset and the different textual features used for grounding the forecasting mechanism. Finally, in Section 4 we present our evaluation methodology and the different results we obtained.

2 Methods

In this work, our main hypothesis is that novelty appears when we are not able to correctly forecast the future. With that idea in mind, we transform our textual content into time series signals and we use a known forecasting algorithm in order to learn the dynamics and predict the evolution of our classification scheme.

2.1 Univariate forecasting method

Since the novelty that we want to detect corresponds to an unexpected change in the volume of a signal, we use forecasting methods to predict what should be the normal

behavior of our signal $[y_1, y_2, ..., y_t, ..., y_N]$. In simple words, we assume that the normal expected behavior at time t can be expressed as :

$$y_{t+1} = h(y_{0,...,t}) + \epsilon_t$$

where ϵ_t is a zero correlation white noise process.

We predict only the next value of our signal y_{t+1} using the past values $y_{0,...,t}$. In this section, we present two forecasting algorithms: the first one is based on K-Nearest Neighbors (Martinez et al., 2019), and the second one is based on ARIMA.

- K-Nearest Neighbors: traditionally used in classification and regression, we can use the KNN algorithm for prediction. Let l be a positive integer representing a maximum lag, then we use the l previous values $y_{t,...,t-l}$ to predict the next value y_{t+1}. Applying a rolling mechanism, we are able to train a model with some features $[y_{t,...,t-l}]$ and targets y_{t+1}.
- ARIMA: AutoRegressive Integrated Moving Average is one of the most popular linear models in time series forecasting. It is based on the assumption that it is possible to forecast a signal using only its past values. It is a linear regression model that uses its own lags as predictors.

2.2 Forecasting using exogenous variables.

We use a forecasting model[3] based on constructed features derived from our initial textual data.

Let $[y_1, y_2, ..., y_t, ..., y_N]$ be a time series where y_t is the value of the time series at time t (e.g., hours, days or months) and $[x_{1f}, x_{2f}, ..., x_{tf}, x_{Nf}]$ be a multivariate time series where x_{tf} is the value of feature f at time t. Our model should be in the form:

$$y_{t+1} = h(x_{t,1}, x_{t,2}, ..., x_{t,F}) + \epsilon_t$$

where F is the number of features used and ϵ_t is an error term. The function h maps an entry $x_t \in \mathbb{R}^F$ to $y \in \mathbb{R}^+$.

We use a Random Forest (Breiman, 2001) to estimate the function h. Although this algorithm is often used for classification, it is also well suited for prediction and time series forecasting (Kumar and Thenmozhi, 2006). Compared to more recent methods based on neural networks, Random Forest has the advantage to give us information about feature importance. It estimates this importance by measuring how much the prediction error increases when data for that feature is permuted while others are left unchanged.

The time period δt is constant: it can be hours, days or months. The random forest model is trained on a subset of the data with $t \in \{1, ..., t_c\}$ with t_c being a constant marking the end of the training window. Once the model is trained, we can use it to

[3] Reproducibility code is available at `https://github.com/clechristophe/CPDPred`

forecast $\hat{y}_{t+1}$ for $t \in \{t_{c+1}, ..., t_N\}$. To this end, we use a rolling mechanism where at each point, we obtain a 1 time-step ahead prediction. The model is evaluated by looking at the error between the prediction and the actual value: $e_{t+1} = y_{t+1} - \hat{y}_{t+1}$. We make the assumption that a large error of prediction is a sign of change. In consequence, we monitor the evolution of e_t to detect out-of-control behavior.

2.3 Cumulative Sum Control Chart

Very popular in statistical processes control, the Cumulative Sum Control Chart (CUSUM) is a sequential analysis method conceived to detect changes on processes. For this, a statistic based on cumulative sums s_t is computed and tracked (Kenett, Zacks, and Amberti, 2013). Deviations from a target value are successively added to get consistent departures of the CUSUM statistic when the process deviates from the target.

A test statistic g_t sums up its input s_t with the idea to raise an alarm when the sum exceeds a threshold h. In order to prevent positive drifts caused by noise in the data, a drift term ν is subtracted at each time-slice. To prevent negative drift, g_t is reset to 0 each time it falls lower than a threshold a. Mathematically,

$$g_t = max(g_{t-1} + s_t - \nu, 0) \text{ or } max(g_{t-1} - s_t - \nu, 0),$$
$$\text{an alarm is raised if } g_t > h$$

In our case, we track the prediction error which should stay in a process of controlled variations since we assume it should be close to white noise. For this, let $\hat{\theta}_t = \frac{1}{t-t_0} \sum e_k$ be the cumulative statistic over the period started just after the last observed alarm t_0. Then, at time t we observe the change on the statistic given by $s_t = e_t - \hat{\theta}_{t-1}$.

CUSUM method depends on two parameters that control the sensitivity to change in the process: threshold h and drift ν. For most applications using CUSUM, values of these parameters can be fixed manually after a thorough analysis of the monitored process. In our case, since we are monitoring several signals at the same time, we compute automatically the values of these parameters. We update our CUSUM parameters based on the maximum values of each of our monitored signal (taken during the training phase): it presents the advantage that raised alerts are not dependent on the different magnitudes of the signals.

$$h = \tfrac{1}{2} max(y_{1,...,t_c}),$$

This threshold value represent the sensitivity to the error of our prediction model. Since we want to limit the number of false alarms and the detection of very small bursts, we fixed this value while experimenting. With the same idea in mind, the drift parameter ν was set to 2.

3 Data and feature importance

3.1 Data

We test our method on two datasets: one public dataset (New York Times Annotated Corpus) and one dataset constituted of anonymized client emails for industrial use for

EDF (*Electricité de France*). Their characteristics are summarized in Table 1.

The New York Times Annotated Corpus (NYTAC)[4] is a public corpus of written articles published by the New York Times between 1987 and 2007. It contains more than 1.8 millions articles that were manually annotated into categories (e.g., Terrorism, Motion Pictures, Politics, Economy). For this work, we focus on the articles published between 1995 and 2005. In order to illustrate our approach, we selected some specific categories to monitor, for example: Terrorism, Basketball, United States International Relations, Motion Pictures, Elections and Colleges and Universities. Figure 1 illustrates the temporal dynamics of these categories in our dataset in terms of number of documents labelled under a category by month. We also selected some categories with noisy and constant behavior in order to properly evaluate our approach. Our dataset contains 13 categories.

dataset	docs	language	# of used cat.	# of used docs	time range	time range for train (t_c)
NYTAC	1.8M	English	13	400k	1995-2005	1995-1998
EDF	100k	French	8	80k	Oct.2018-Oct.2019	Oct.2018-Jan.2019

Table 1. Summary of datasets used for this work

Electricité de France (EDF) is the main french electricity producer. With more than 35 millions clients, EDF receives hundreds of thousands emails per month. In order to best process this data, the business entity have implemented classification methods based on classification schemes constructed by business experts. However, they noticed that the volume of emails associated with each category may vary unusually over time. From a business point of view, it is essential to have a system able to raise alerts when the volume of documents in a category is evolving unusually. The EDF emails corpus is a private and anonymized (in accord with GDPR regulations) corpus of email, in french, sent by clients to EDF between October 2018 and October 2019. EDF classification scheme is organized between 12 main categories but only 8 are exploited in this work because some are unusable: due to very low volume, empty email or containing spam.

[4] `https://catalog.ldc.upenn.edu/LDC2008T19`

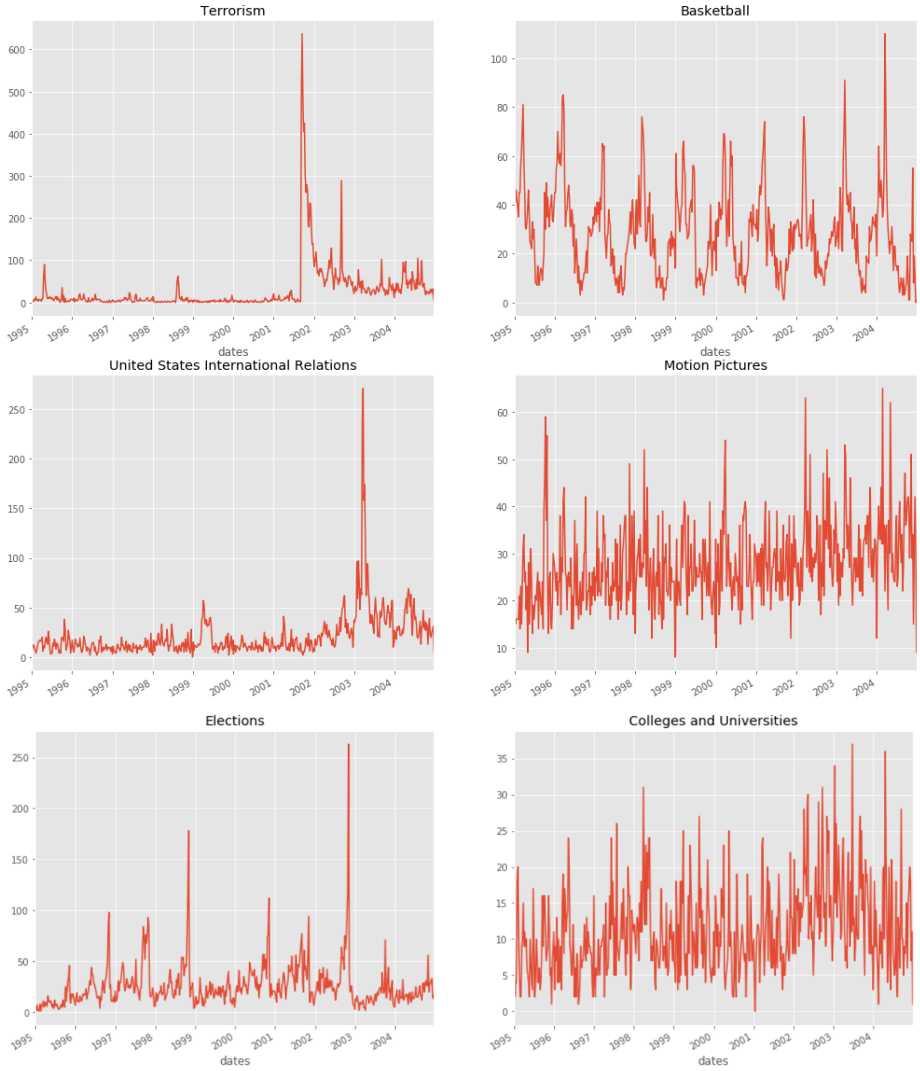

Fig. 1. Evolution of the number of articles published under different categories per day in the NYTAC.

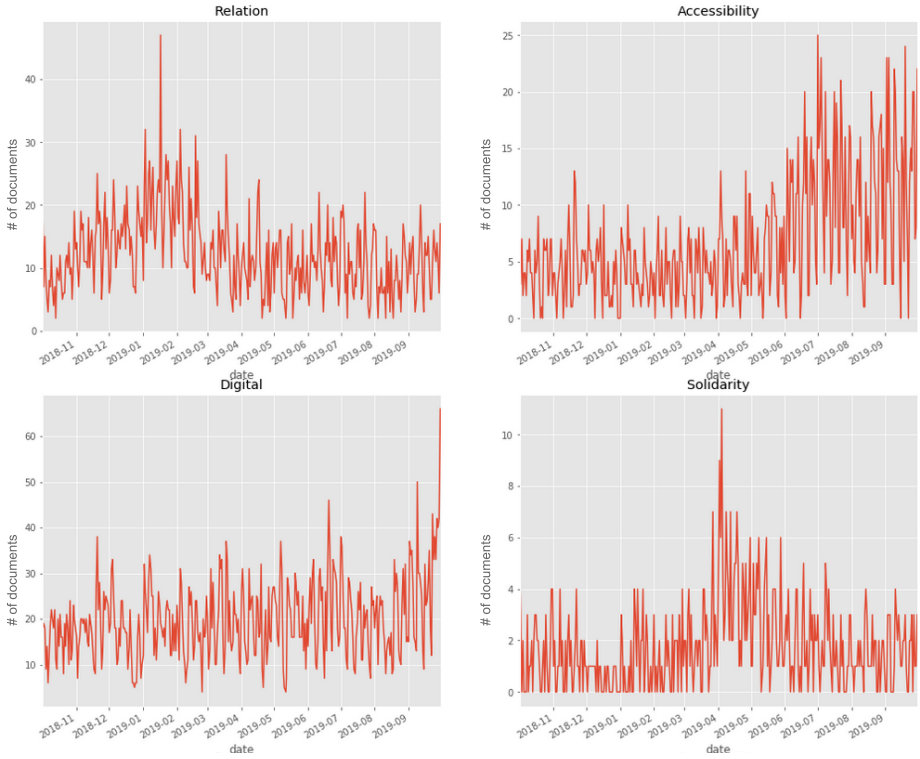

Fig. 2. Evolution of the number of emails sent under different categories per day in the EDF corpus.

3.2 Exogenous variables

We now describe three families of features used in our prediction model:

Frequency of words: we count the occurrences of certain words each day. Since we work with pre-defined categories, we are able to select automatically a certain number of words closely related to one category. For each category, we select a certain number of words to monitor. In addition to the raw frequency of the words, we consider the lag of the frequency $\Delta^p f_t^w = f_t^w - f_{t-p}^w$. We chose to work with $p = 7$ or $p = 365$ to represent the weekly and yearly seasonality. In the EDF dataset, since we only have one year of data, we work only with a weekly lag of $p = 7$. These values of lags are illustrated on Figure 3, we see that the *Basketball* category presents 10 periods over 3650 data points (yearly seasonality) and *Motion Pictures* category present 7 periods over 49 data points (weekly seasonality).

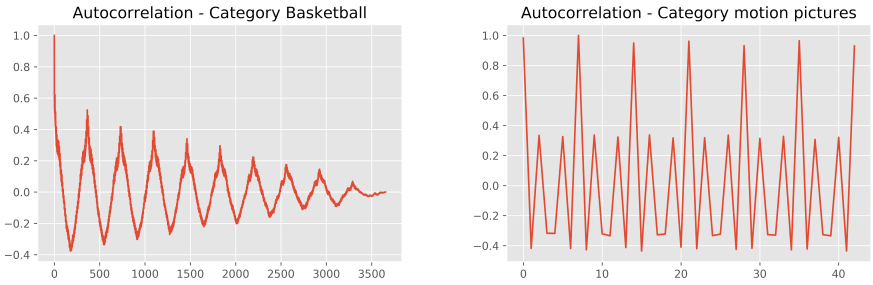

Fig. 3. Auto-correlation for 2 categories of the NYTAC. We see an yearly (left) and weekly (right) seasonality.

Frequency of topics: Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan, 2003) is a topic modeling algorithms that seeks to discover topic structures hidden in a textual dataset. Using probabilistic generative model, it describes each document as a mixture of topics θ^d and each topic as a distribution over the vocabulary ϕ_z. In this work, we train a LDA model on our train dataset $D_{train} = D_0, D_1, ...D_{t_c}$ with 30 and 20 topics respectively for the NYTAC and EDF dataset with $|D_t|$ being the number of document published at time t. We form signals corresponding to the number of times a given topic occurs each day. For each document arriving in our corpus in a streaming manner, we infer its mixture of topic by observing the words that compose it. We have a time series:

$$[k_{1,i}, ..., k_{t,i}, ..., k_{T,i}] \text{ and } k_{t,i} = \frac{1}{|D_t|} \sum_{j=1}^{|D_t|} p(z_i|d_{j,t})$$

With z_i being ith topic and d_j a document in D_t. We also represent this time series with lags values at $p = 7$ and $p = 365$.

Table 2 shows the 5 most probable words associated with some topics built with LDA and Figure 4 shows the frequency of these topics over the entire corpus of the *New York Times*. We observe that information is carried by the words associated with each topic but also by its temporal aspect. For example, we notice a change of behavior over time in topic 17, as well as a cyclic pattern in topics 7 and 21 that can give us information about the temporal dynamics of our data. We will see in Section 4 that these unsupervised topics behaviors are useful to predict the evolution of our classification scheme.

Frequency of co-occurrence of topics: for each topic pair $z_{i,j}$ we count the number of times they occur in the same documents at time t. A topic z_i is considered as present in a document if $p(z_i|d) > 0.1$. This threshold is chosen to maximize the impact of these features on forecasting.

Basketball	Justice	College Sports	Cinema	Art	US President
Game	Case	Tournament	Film	Work	State
Point	Court	College	Movie	Show	United
Team	Lawyer	Connecticut	Director	Art	American
Knicks	Judge	State	Hollywood	Artist	Clinton
Second	Trial	National	Star	Image	President
Topic 21	**Topic 12**	**Topic 7**	**Topic 29**	**Topic 6**	**Topic 17**

Table 2. Top words associated with some topics on the NYT dataset.

Access	Consumption Monitoring	Communication	Invoices	Contract change
Fil	Equipe	Pièce	Compte	Contrat
Consommation	e.quilibre	Jointe	Suite	Demande
Acces	Consommation	Trouver	Facture	Service
Jour	Electricité	Téléphone	Service	Résiliation
Actualité	Distribuer	Ci-joint	Réclamation	Offres
Topic 8	**Topic 11**	**Topic 7**	**Topic 5**	**Topic 15**

Table 3. Top words associated with some topics on the EDF dataset.

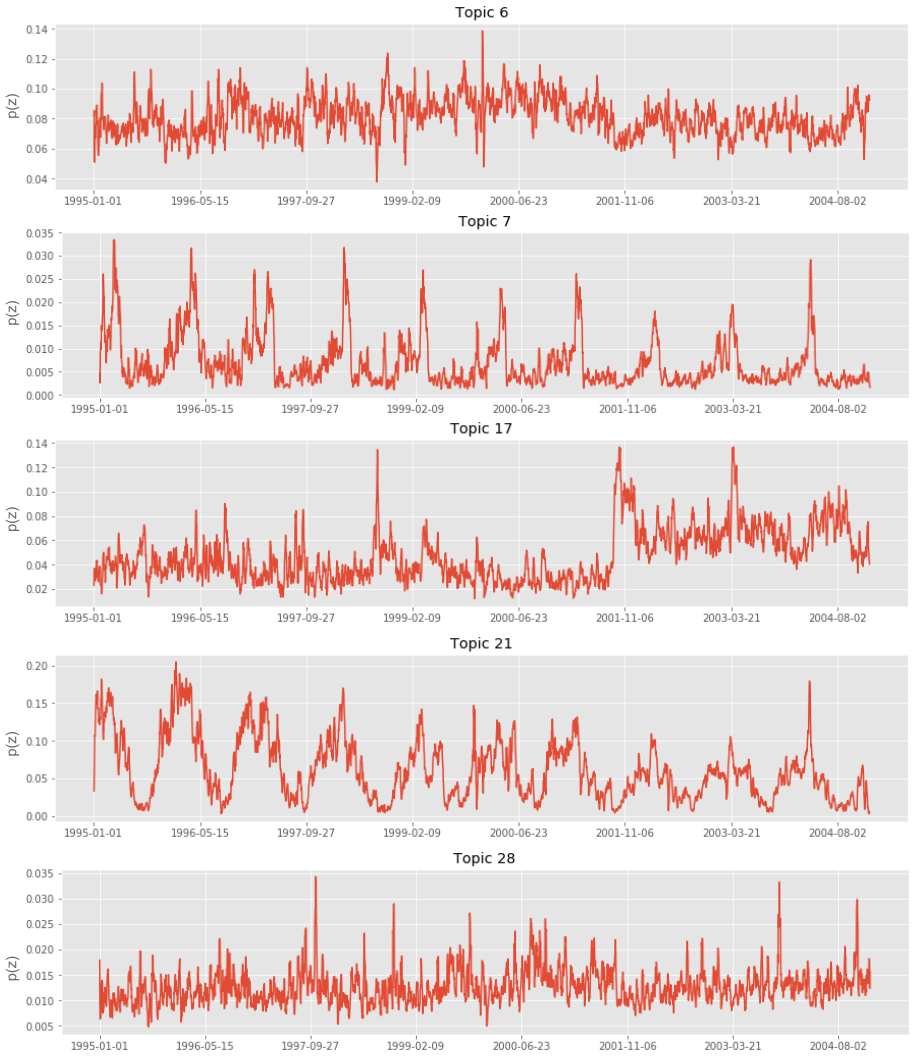

Fig. 4. Evolution of the frequency of some topics over time on the *New York Times* dataset.

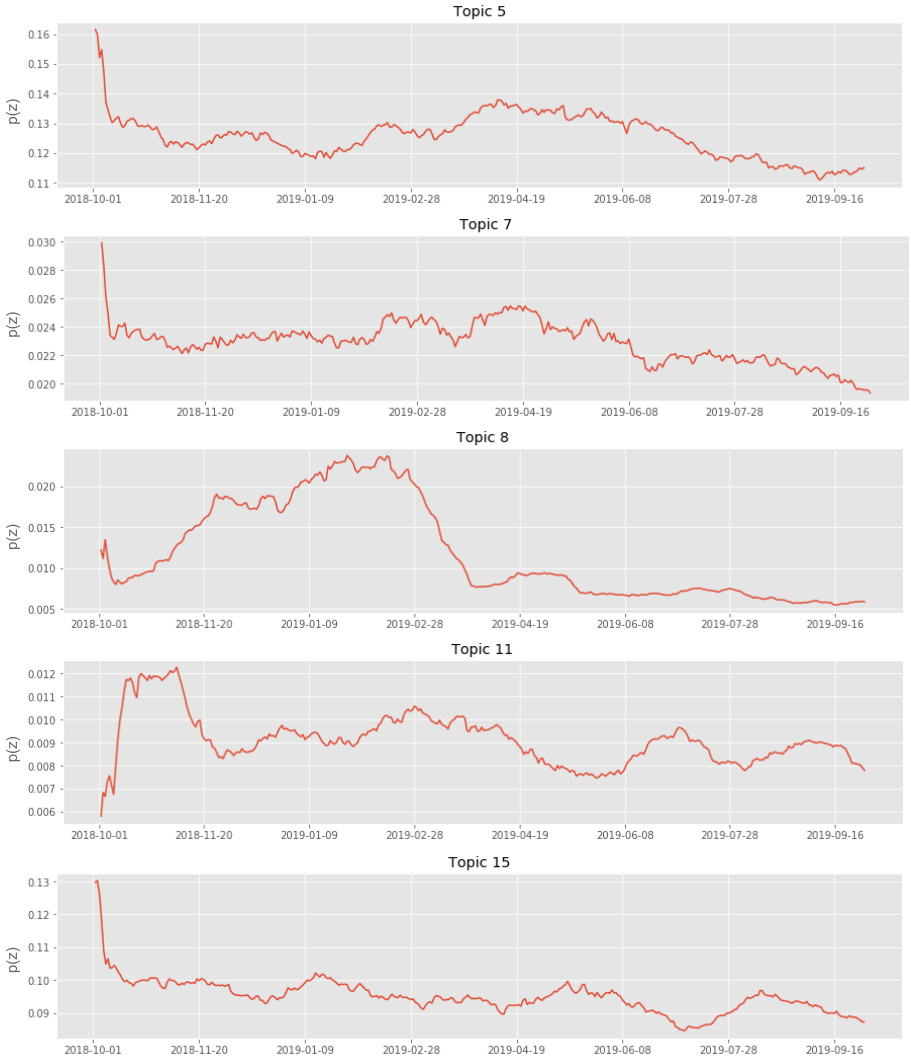

Fig. 5. Evolution of the frequency of some topics over time on the EDF dataset.

3.3 Feature selection

In Section 3.2, we saw that we select a certain number of features to include in our forecasting model. Indeed, since it is computationally expensive to monitor the frequency of the entire vocabulary, we choose to automatically select a few unique words that describe effectively the category we are monitoring.

In order to select which words to monitor for each category in the NYT, we solve a classification problem with a Naive Bayes classifier on the training part of our data

(from 0 to t_c). We represent each word in a term frequency-inverse document frequency (TF-IDF) space. TF-IDF for a term t of a document d in our document set is computed as follows:

$$tfidf(t,d) = tf(t,d) * idf(t)$$
$$idf(t) = log[\frac{n}{df(t)}] + 1$$

where n is the total number of documents in the document set, $tf(d,t)$ the frequency of term t in document d and $df(t)$ is the document frequency of t; the document frequency is the number of documents in the document set that contain the term t. While this classifier obtains 87% of accuracy on the New York Times, we are more interested at observing the most discriminative features for each class. Some of these are provided as examples in Table 4.We chose to solve our classification problem with a Naive Bayes classifier on a TF-IDF matrix because it is simple, fast to compute and has good performance over our two datasets and is also easier to generalize to other datasets and other languages.

Categories presented in the EDF dataset are more specific. While it would be possible to apply the same technique to select words to monitor, we preferred to use the internal expertise of the company to determine the important features. We asked some business experts to manually select some words to monitor for each category.

Terrorism	**Art**	**Motion Pictures**	**Basketball**	**US Itl. Relations**
attack	art	film	game	clinton
terrorist	museum	movie	knicks	united
state	painting	actor	team	state
official	artist	director	point	president
federal	work	hollywood	player	american
american	gallery	directed	basketball	official
people	show	story	season	nato
united	exhibition	character	net	palestinian
bombing	sculpture	theater	coach	administration
terrorism	new	festival	play	china

Table 4. Most discriminative words for some categories of the *New York Times*

4 Experiments and results

In this Section, we measure the contribution of the exogenous variables to the forecasting algorithm. We show how the different features contribute to the forecast and explain them. We present the results obtained by our model CDPred (for Change Detection with Prediction) in terms of consistence of alerts: we want a model able to raise alerts at the right time. We explore the timeline of alerts raised and link them with real events on the *New York Times* dataset. We compare the sensitivity to change and the amount of

change needed to raise an alert between CDPred and some baselines. Finally we present the alerts raised in an industrial context and how it can be useful to EDF.

4.1 Contribution of the exogenous variables

In this Section, we present the forecasting performance of the models presented in Section 2. Our main hypothesis is that some observable variables in the textual content of our data are useful to accurately forecast the evolution of a signal (i.e. the volume of documents classified into a category). We compare the Root Mean Square Error (RMSE) of our different forecasting models:

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{N}}$$

with $\hat{y}_i$ being the predicted value of the time series at time i, y_i the true value, and N the length of the time series.

Since we have one model per category, we compute the mean of all our RMSE for the whole dataset. RMSE could hide different behavior between signals of each categories as they have different magnitudes. We observed that the difference in RMSE between models is overall proportional.

Results are presented in Table 5. We observe an overall better prediction using a Random Forest Model using exogenous variables. It is important to note that the best combination of parameters have been used for each forecasting model. For example, the random forest model used for forecasting has a hyperparameter n corresponding to the number of trees in the forest. Here $n = 500$ has been chosen using a k-fold cross validation ($k = 10$). Since the Random Forest is the best model for prediction, we will present the results (in terms of alerts raised) using this model.

Algorithm	Mean RMSE on NYTAC	Mean RMSE on EDF
KNN-Prediction	4.15 ± 1.22	13.08 ± 1.88
ARIMA	3.97 ± 1.08	10.41 ± 1.52
Random-Forest	2.41 ± 0.82	7.87 ± 1.36

Table 5. RMSE of different prediction algorithms. KNN-Prediction and ARIMA are univariate and Random-Forest is using exogenous variables described in Section 3.

4.2 Feature importance

First of all, as we use a Random Forest model for prediction, we can observe the feature importance for the forecasting of each category. Table 6 shows some features importance for 5 categories of the *New York Times*. The words associated with the topic shown in this Table are presented in Table 2. Table 7 shows the features importance for

4 categories of the EDF dataset.

For most of the monitored categories, frequencies of certain words are more important for the forecast. For the New York Times, the 6 most useful features about the *Terrorism* category are words frequencies and nearly all are about the words "Bombing" and "Federal". Indeed, the raw frequency brings information and the use of lags is also useful for the forecasting. These lags carry information about the change in periodicity of the time series and seem to be helpful in case of sudden peak in the data. We will discuss this hypothesis later. In the EDF dataset, we observe that the *Digital* category is best described with the help of 4 words frequencies.

In some categories, for example *Motion Pictures*, *Art* and, most importantly *US Itl. Relations*, information is contained in the LDA topic signals. Indeed, its main feature is the signal corresponding to the evolution of Topic 17. In Table 2, Topic 17 is about US President and its top words are about the United States and the American President Clinton. It is a good description for a *US Itl. Relations* category in the *New York Times*.

Also, we observe that for categories *Basketball* and *Relation*, the co-occurrence signal of 2 topics brings some information for the forecasting algorithm. For *Basketball*, Topic 7 and 12 are, respectively about College Sports and Justice. While the first is rather close to basketball in term of words used, we could argue that the second brings information because of the word "Court" having a meaning in both basketball and justice context and because of several justice cases around the NBA (National Basketball Association) in the early 2000s.

Terrorism	**Basketball**	**Motion Pictures**	**Art**	**US Itl. Relations**
bombing	game	film	art	Topic 17
bombing_365	basketball	directed	gallery	Clinton
federal_365	Topic 21	Topic 29	Topic 6	united
federal	Cooc 12 & 7	Topic 1	painting_365	united_365
federal_7	team	movie_365	gallery_365	president_365
terrorist	coach	story	art_365	state_365

Table 6. Top features used for each category forecast on the *New York Times*. "_365" and "_7" indicate, respectively, the use of the lag $p = 365$ and $p = 7$."Cooc" indicates the Frequency of co-occurrence of some topics

4.3 Baselines

In order to evaluate our model CDPred, we compare it to two baseline from the literature. Our first baseline is adapted from Allan et al., 2000 (TF-IDF) where a method to detect and track new topics is presented. This method is based on the popular term frequency–inverse document frequency (TF-IDF) statistic and is built to raise alerts on

Relation	Accessibility	Digital	Solidarity
connecter	téléphone	mail	social
Topic 8	joindre	site	Topic 5
Cooc Topics 11 & 6	numéro	numéro	Topic 15
compte	Topic 7	internet	assistant

Table 7. Top features used for each category forecast on the EDF dataset. "Cooc" indicates the Frequency of co-occurrence of some topics

particular terms when their TF-IDF statistics cross a manually-determined threshold. The second algorithm is Xie et al., 2016 (TopicSketch). It is an algorithm based on the monitoring of physical measurements (speed, acceleration) of textual entities (words and n-grams). It is built to raise alert when these statistics have crossed a threshold.

In order to compare to CDPred that raises alerts not on specific words but rather on a signal, we selected alerts raised on words that are clearly linked with our categories. We selected 100 words per category with the method presented in Section 3.3. Selected words seem to discriminate as much qualitatively (Table 4) as quantitatively (Table 6)

In Xie et al., 2016 (TopicSketch) and Allan et al., 2000 (TF-IDF), threshold values are set manually. We ran experiments with different values of threshold and we present the optimal results in terms of number of alerts raised. It means that we are minimizing the ratio between number of interesting and false alerts. These are presented in Figure 6.

We clearly see that for Allan et al., 2000 (TF-IDF), alerts are raised randomly compared to the observed signal. We notice a great number of alerts around the peak in *US Itl. Relations* but other than that, we can conclude that this method is not effective to resolve our task.

Xie et al., 2016 (TopicSketch) seems to raise false alarms in categories such as *Theater* and *Art*. Alerts raised in *Basketball* and *US Itl. Relations* make sense compared with the dynamic of the signal. In *Basketball*, Xie et al., 2016 (TopicSketch) has a tendency to raise alerts at each peak of the season. In *US Itl. Relations*, we notice that it raises alerts at each small peak in the volume of documents classified under this category. While it seems to be a good solution to detect sudden bursts in the data, it is not a good method to detect long-term changes in the dynamic. In other words, it has no memory of the usual dynamic of the monitored signal.

4.4 Alerts raised on the *New York Times*.

In Figure 7, we can see the alerts raised by our model in 6 categories annotated in the *New York Times*: *Terrorism*, *US International Relations*, *Basketball* and *Politics and Government*, *Elections*, *Colleges and Universities*. These categories are studied here because they are the ones that present an interesting temporal aspect. *Terrorism*, *US International Relations* and *Elections* categories are linked with major events that conducted in massive changes in the editorial slant of the *New York Times* and obviously

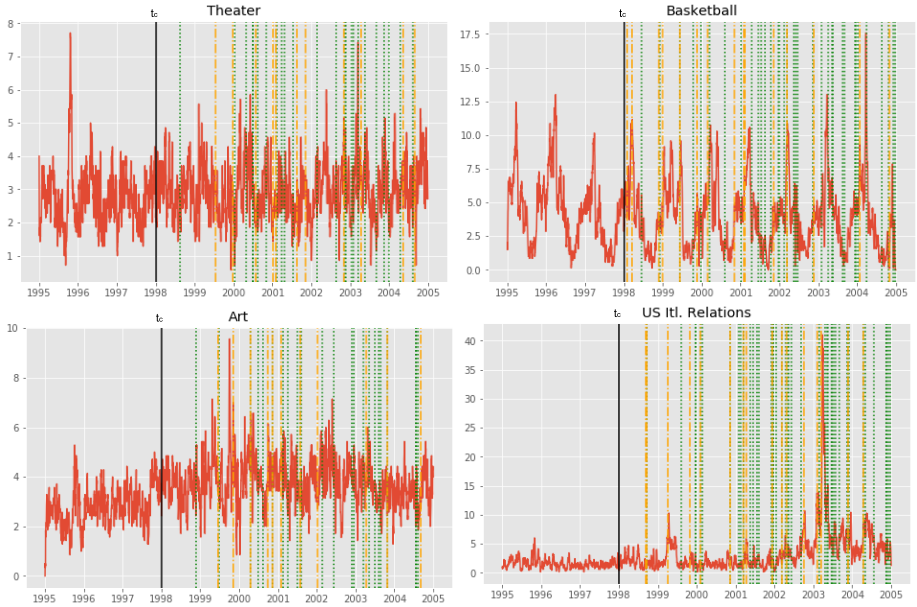

Fig. 6. Alerts raised by our baselines TF-IDF (Allan et al., 2000) (dotted green) and TopicSketch (Xie et al., 2016) (orange) on 4 categories of the NYTAC

in the temporal signal of the annotated categories. In this three categories, we can identify five bursts that can be considered as unexpected compared to what we know of the evolution of the category:

- Burst 1. August 21st 1998: the bombing of targets in Afghanistan and Sudan by the US military had the effect of tightening the security of American cities and particularly New York City.
- Burst 2. September 11th 2001: terrorist attacks on the *World Trade Center*.
- Burst 3. March and April 1999: entry of NATO forces into the Kosovo war.
- Burst 4. March 20th 2003: invasion of Iraq by the US military
- Burst 5. November 5th 2002: midterms elections in the US.

For bursts 2 and 4, we see that our alerts raised by CDPred are well localized around these events. For *US International Relations*, compared to alert raised by our baselines in Figure 6, we see that our alerts are much more consistent with real world events. For burst 3, even if the peak is much lower than burst 4, it has considerably changed the proportion of articles tagged under the *US International Relations* category for a duration of 2 month. We see that our system has correctly detected this burst very early. For burst 1, it has provoked only a one-day surge in the number of article tagged under the *Terrorism* category, we could reasonably argue that it is a false alarm, as it do not correspond to more long term change in the classification. However, it also corresponds to a very abnormal peak in the volume of document under this category. For burst 5, we see that, even if elections correspond to a cyclic signal, the peak around November

2002 is abnormal in terms of volume and start earlier than usual.

Categories about *Basketball* and *Politics and Government* are also interesting to monitor. The *Basketball* signal is cyclic, as it follow the NBA and NCAA season in the United States. We see, in Figure 6, that our TopicSketch baseline has a tendency to raise an alert every time a peak in present in the signal, even if this peak happen every year. On the other hand, we see that our model CDPred did not raise any alert for this category. We can conclude that it has correctly learned the correct dynamic of the category. The *Politics and Government* category seem to be fairly constant but also very noisy over time. By looking at the signal, we could hardly argue that a change has occurred in the dynamic of the category. However, CDPred has raised an alert at the following date: 2nd January 2002. This date corresponds to the day of the inauguration of the new mayor of New York City Michael Bloomberg.

In Figure 8, we present other categories annotated in the *New York Times*: *Art*, *Theater*, *Automobiles* and *Dancing*. These categories all have a fairly constant but noisy signal: no changes in the dynamics are observed in these categories. We see that, compared to alerts raised by our baselines presented in Figure 6, our model CDPred did not raise a lot of alerts. Only 4, that can be considered as false alarm, have been raised by CDPred on these 4 categories.

In Table 8 we can observe the difference between the date of the events detected by CDPred in 7 and the date of the first alert raised by CDPred compared to our baselines. We have $\Delta_{method} = t_{change} - t_{alert}$. We notice that, for very big bursts, like the *WTC terrorist attacks* and the start of Iraq War, the three model are able to raise alerts fairly quickly. However, our model CDPred detect the unexpected change in the volume faster than the others. We saw that the entry of NATO in the Kosovo War had an long term impact on the dynamic of the *US International Relations* category and we observe in this table that CDPred was faster that TopicSketch to detect the change and that the baseline TF-IDF never detected it.

Category	Event	t_{change}	Δ_{tfidf}	Δ_{Sketch}	Δ_{CDPred}
Terrorism	Security tightening	1998-08-21	33	19	**0**
Terrorism	*WTC* terrorist attacks	2001-09-11	3	1	**1**
US Itl. Relations	Kosovo War	1999-03-23	251	13	**10**
US Itl. Relations	Iraq War	2003-03-20	12	5	**0**
Elections	Mid Terms Elections	2002-08-30	14	11	**7**

Table 8. Absolute value of the Δ (in days) between the actual day of the change and the day of the alert.*Sketch* corresponds to Xie et al., 2016 and *tfidf* corresponds to Allan et al., 2000.

Finally, we see in Figure 8 that some categories do not seem to present any change in their dynamics, they are fairly constant so they should not present any alerts raised during our monitoring. We present in Table 9, the number of alerts raised in these cate-

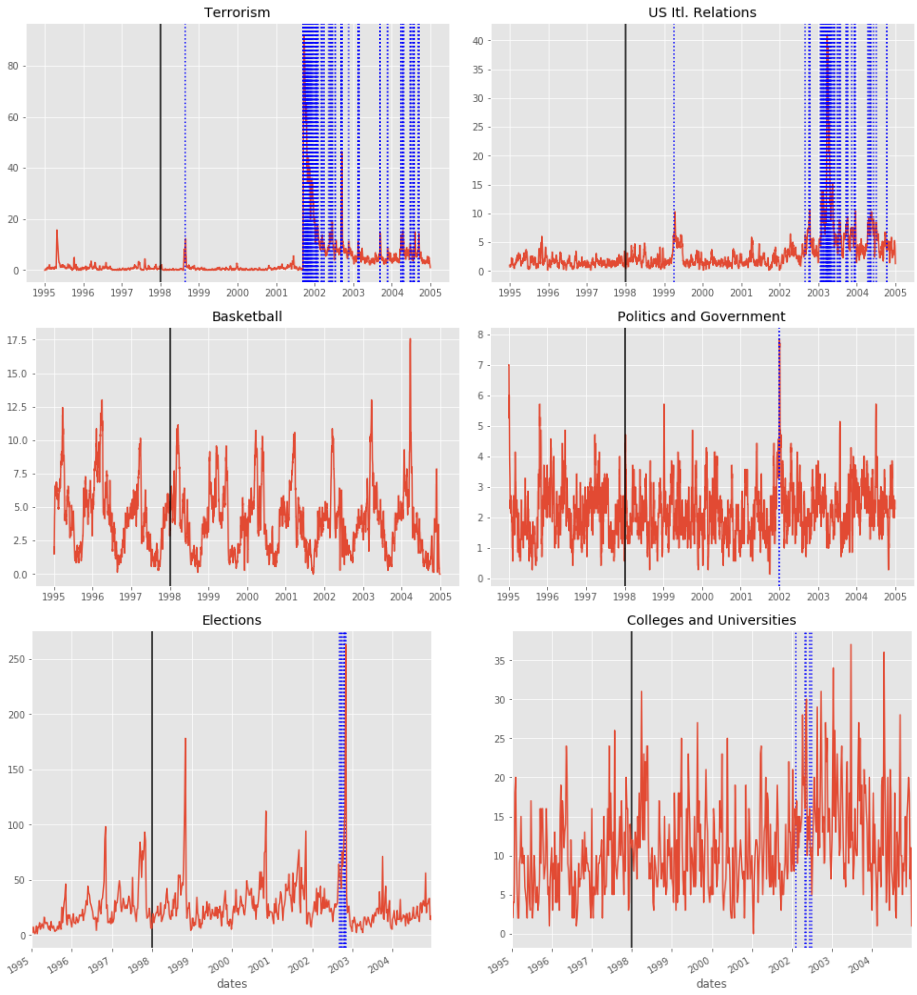

Fig. 7. Alerts raised by CDPred (dotted blue) on 6 categories of the NYTAC

gories by our model CDPred and the other baselines. These alerts are considered false alerts as they do not correspond to real major events that have been identified or to any changes of dynamics. We see that our model is less likely that tfidf Allan et al., 2000 and TopicSketch Xie et al., 2016 to raise false alerts on these constant categories.

In this Section, we observed the results of our model CDPred on different tasks. While it is difficult to evaluate quantitatively this type of model, we saw that our approach is efficient to detect bursts that have a long-term impact on a signal. We saw that CDPred is less prone to raise alerts on one-time burst, and to raise false alarm when the dynamic of the signal is not changing. Also, CDPred seems to be faster than the

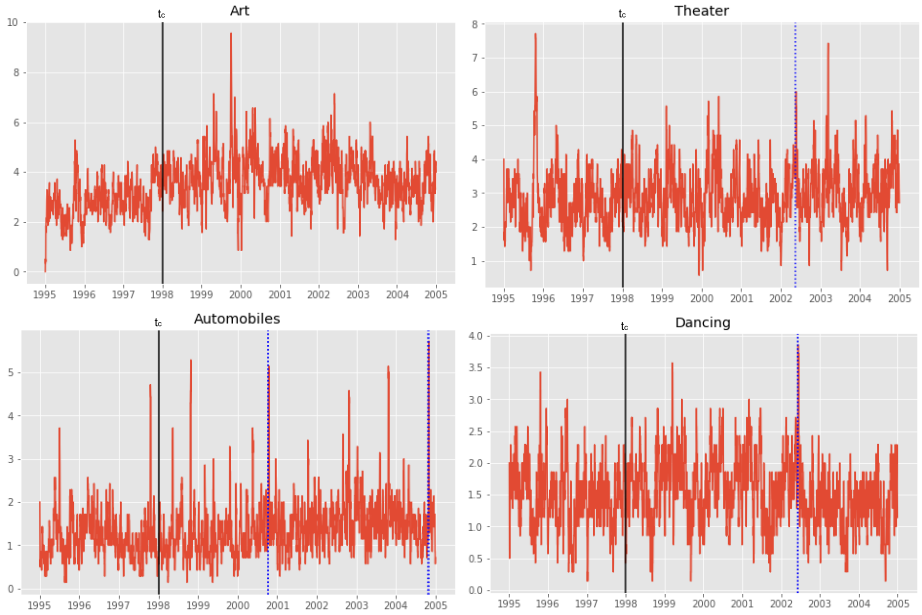

Fig. 8. Alerts raised by CDPred (dotted blue) on 4 categories of the NYTAC

baseline to raise an alert. Finally, CDPred present the benefit of a long-term memory of the signal. It means that it will not raise alert for cyclic signals.

4.5 Industrial application

We mentioned in Section 1 that the main motivation behind this work was to be able to raise early alerts when changes in the dynamics of a classification scheme are detected. The CDPred model has been studied in an industrial context and we will present in this Section the main results on the EDF dataset, presented earlier.

In Figure 9, we observe the alerts raised for 4 categories: *Relation with EDF*, *Accessibility*, *Digital* and *Solidarity*.

Categories	N_{Sketch}	N_{tfidf}	N_{CDPred}
Art	10	32	0
Murders	6	10	0
Theater	11	24	1
Travel	4	18	1
Dancing	2	8	1
Automobiles	5	10	2

Table 9. Number of false alerts N raised by each model for different constant categories. *Sketch* corresponds to Xie et al., 2016 and *tfidf* corresponds to Allan et al., 2000.

- For *Accessibility*, we see that the change of dynamic that has begun around 2019-06 is clearly detected by our model and that a lot of alerts are raised after that because the new dynamic still do not correspond to the normal dynamic observed during training time.
- For *Solidarity*, we observe a peak occurring for about two months (between 2019-04 and 2019-06) that is also clearly detected by our model. It is interesting to note that this peak is detected even if the normal volume of documents classified under this category is very low: about one or two documents per day.
- For *Relation with EDF*, the unusual volume seems to appear at the beginning of our observation (even during training time). We see that the end of this burst is detected at the beginning of 2019-01.
- Finally, for *Digital*, three alerts are raised, one in 2019-06 that does not seem to correspond to any change in the dynamic and two other that are occurring at a sudden peak in the volume of document. These alerts may be considered as false alarms. However, we can see that the change of dynamic seems to appear at the very end of our observation and it is legitimate to wonder whether, with a few more observations, we could have detected this change.

Compared to the New York Times, the EDF dataset has less one-time burst and some categories contains real changes of dynamics during our period of observation. We saw that CDPred is able to detect this changes and, since it is a private corpus, we are not able to discuss the reason behind these changes.

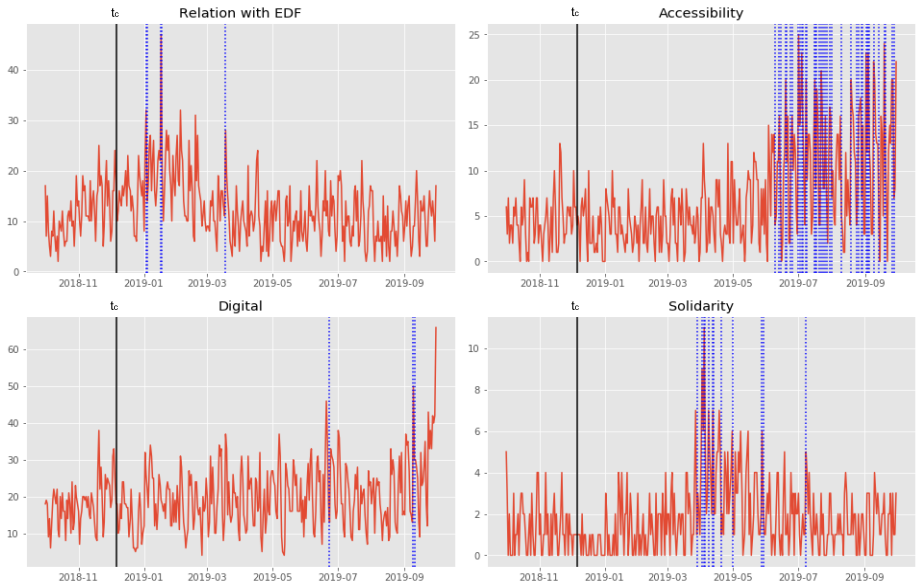

Fig. 9. Alerts raised by CDPred (dotted blue) on 4 categories of the EDF Dataset

5 Conclusion and future work

In this work, we presented an algorithm capable of raising alerts of unexpected change in a stream of textual documents. To this end, we cast our textual content in the form of a temporal signal and we observed the changes in the volume of a category. We based our method on the hypothesis that an unexpected change in a signal is tied to a greater forecasting error. Instead of using traditional endogenous forecasting method directly on the signal, we used exogenous variables observed in the textual content of our dataset. Using classical textual analysis and topic modeling techniques to build these features, we showed that we obtain a better forecasting model without adding much more complexity. The goal of this work was not to obtain a perfect forecasting model but rather having a model with information about the underlying signal it tries to predict. Random-Forest algorithm for prediction has been chosen due to its great explicability in relation with the textual features.

Then we combined this forecasting model with a sequential analysis method, traditionally used in the monitoring of industrial process. By applying the CUSUM algorithm on our prediction error, we are able to evaluate its stability and to raise alerts as soon as it is considered unusual. We demonstrated that our model is good at noticing large bursts and also better than the well-established methods chosen as baselines for small bursts. By being based on a forecasting algorithm, our model present the advantage of not being sensitive to cyclicity. Indeed, we showed that, compared to the baselines, it does not raise any alerts when the signal is cyclic. Finally, CDPred is less sensitive to noise in the sense that it raises less false alarms on one-day events and on constant categories.

In terms of textual features, it may be interesting to study other types of information, such as the number of verbs, nouns and adjectives since it should differ from one category to another. Another interesting feature to represent our textual entities could lie in the use of embeddings models such as Word2Vec (Mikolov et al., 2013). For our forecasting model, we chose a simple Random Forest because we kept in mind that one of the main extension of this work could be on the explicability of the alerts. We can analyze the importance of each feature to identify the words or documents responsible for an alert. Also, a good extension to the model would reside in the possibility to validate an alert in order to avoid large group of alerts concerning the same event (e.g. categories *Terrorism*, *US Itl. Relations* and *Accessibility*. Finally, since we observed the change in the dynamic of a textual data stream and we developed an explainable model concerning this change, we suppose that it would be possible to learn to detect the weak signals appearing before the change.

References

Allan, James et al. (2000). “Detections, bounds, and timelines: Umass and tdt-3”. In: *Proceedings of topic detection and tracking workshop*. sn, pp. 167–174.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Breiman, Leo (2001). “Random forests”. In: *Machine learning* 45.1, pp. 5–32.

Christophe, Clément et al. (2019). “How to detect novelty in textual data streams? A comparative study of existing methods”. In: *4th ECML PKDD Workshop, AALTD 2019, Würzburg, Germany, September 20, 2019, Revised Selected Papers*.

Huang, Jiajia, Min Peng, and Hua Wang (2015). “Topic detection from large scale of microblog stream with high utility pattern clustering”. In: *Proceedings of the 8th Workshop on Ph. D. Workshop in Information and Knowledge Management*, pp. 3–10.

Kenett, Ron S, Shelemyahu Zacks, and Daniele Amberti (2013). *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. John Wiley & Sons.

Kumar, Manish and M Thenmozhi (2006). “Forecasting stock index movement: A comparison of support vector machines and random forest”. In: *Indian institute of capital markets 9th capital markets conference paper*.

Lau, Jey Han, Nigel Collier, and Timothy Baldwin (2012). “On-line trend analysis with topic models:\# twitter trends detection topic model online”. In: *Proceedings of COLING 2012*, pp. 1519–1534.

Lazzaretti, André Eugênio et al. (2016). “Novelty detection and multi-class classification in power distribution voltage waveforms”. In: *Expert Systems with Applications* 45, pp. 322–330.

Long, Rui et al. (2011). “Towards effective event detection, tracking and summarization on microblog data”. In: *International Conference on Web-Age Information Management*. Springer, pp. 652–663.

MacGregor, John F and Theodora Kourti (1995). “Statistical process control of multivariate processes”. In: *Control Engineering Practice* 3.3, pp. 403–414.

Markou, Markos and Sameer Singh (2003). “Novelty detection: a review—part 1: statistical approaches”. In: *Signal processing* 83.12, pp. 2481–2497.

Martinez, Francisco et al. (2019). “Time Series Forecasting with KNN in R: the tsfknn Package”. In: *The R Journal*.

Metzler, Donald, Congxing Cai, and Eduard Hovy (2012). “Structured event retrieval over microblog archives”. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 646–655.

Mikolov, Tomas et al. (2013). “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*, pp. 3111–3119.

Noyez, Luc (2009). “Control charts, Cusum techniques and funnel plots. A review of methods for monitoring performance in healthcare”. In: *Interactive cardiovascular and thoracic surgery* 9.3, pp. 494–499.

Page, Ewan S (1954). “Continuous inspection schemes”. In: *Biometrika* 41.1/2, pp. 100–115.

Peng, Min et al. (2018). “Emerging topic detection from microblog streams based on emerging pattern mining”. In: *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*. IEEE, pp. 259–264.

Pimentel, Marco AF et al. (2014). “A review of novelty detection”. In: *Signal Processing* 99, pp. 215–249.

El-Shal, Shendy M. and Alan S Morris (2000). “A fuzzy expert system for fault detection in statistical process control of industrial processes”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 30.2, pp. 281–289.

Tartakovsky, Alexander G, Aleksey S Polunchenko, and Grigory Sokolov (2012). “Efficient computer network anomaly detection by changepoint detection methods”. In: *IEEE Journal of Selected Topics in Signal Processing* 7.1, pp. 4–11.

Tsai, Flora S et al. (2011). “Multilingual novelty detection”. In: *Expert Systems with Applications* 38.1, pp. 652–658.

Xie, Wei et al. (2016). “Topicsketch: Real-time bursty topic detection from twitter”. In: *IEEE Transactions on Knowledge and Data Engineering* 28.8, pp. 2216–2229.