Learning *When* to Classify for Early Text Classification

Juan Martín Loyola[1,2], Marcelo Luis Errecalde[1](✉), Hugo Jair Escalante[3], and Manuel Montes y Gomez[3]

[1] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional, San Luis, Argentina
✉ merreca@unsl.edu.ar
[2] Instituto de Matemática Aplicada San Luis, San Luis, Argentina
[3] Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México

Abstract. The problem of classification is a widely studied one in supervised learning. Nonetheless, there are scenarios that received little attention despite its applicability. One of such scenarios is *early text classification*, where one needs to know the category of a document as soon as possible. The importance of this variant of the classification problem is evident in tasks like sexual predator detection, where one wants to identify an offender as early as possible. This paper presents a framework for early text classification which highlights the two main pieces involved in this problem: classification with *partial information* and deciding the *moment* of classification. In this context, a novel approach that learns the second component (*when* to classify) and an adaptation of a temporal measurement for multi-class problems are introduced. Results with a classical text classification corpus in comparison against a model that reads the entire documents confirm the feasibility of our approach.

Keywords: Early text classification; classification with partial information; decision of the moment of classification

1 Introduction

Recent years have shown a tremendous growth in the machine learning field, solving very complex tasks with new algorithms, methods or architectures [5]. There are, however, settings of the classification problem that have received little attention despite its wide applicability. One of such scenarios is that of *early text classification* (ETC), which deals with the development of predictive models that can determine the class a document belongs to as soon as possible. Here a document is assumed to be processed sequentially, starting at the beginning and reading its containing parts one by one. In this context, it is desired to make predictions with as little information (as soon) as possible.

To date, only a few papers have approached this kind of scenarios [3, 4, 6]. Despite its low popularity, this topic has a major potential in practical applications. For instance, consider the problem of detecting sexual predators in chat conversations. Here, the goal is to sequentially read a conversation and to determine as

fast as possible whenever a sexual predator is involved; clearly, a detection using the whole conversation can only be used for forensics rather than for prevention. Other potential applications include the analysis of conversations that requires of a fast response (e.g., cyber-bullying prevention, detection of early traces of depression, suicidal speech identification) and classification processes where late classification implies some type of cost (e.g., a real-time system where one might need to classify a document without processing it completely to give the user a fast response, otherwise the user might leave the site).

It is important to note that the early text classification problem consists of two related and complementary tasks. On the one hand, the task of *classification with partial information* (CPI), which consists of obtaining an efficient predictive model when only partial information is available that has been read sequentially up to a certain point in time. The emphasis in this case is to determine which classification methods are more likely to achieve performance comparable to that obtained when classified using the entire document. On the other hand, we have the task of *decision of the moment of classification* (DMC), that is, in which point in time one can stop reading and classify with some degree of confidence that the prediction is going to be correct. Both tasks need to be consistently integrated into any system for the ETC problem. However, as we will see in the related work section, little efforts have been dedicated to comprehensive approaches that simultaneously address them.

This paper addresses that previous research gap by presenting a simple framework for ETC which explicitly models the CPI and DMC components. In our proposal, the CPI component is learned with standard machine learning algorithms as in other works in ETC. The novelty of our approach consists in also learning the DMC component given an initial dataset. Evaluations of the ETC systems were carried out with standard classification measures and also with others that take into account the time dimension. In this context, another contribution of our work is the adaptation of a previous temporal evaluation metric for multi-class classification. Experimental results of our approach in a classical text classification corpus show the feasibility of the proposal for the ETC task.

The remainder of this paper is organized as follows. Next section reviews related work on early text classification. Then, Section 3 describes the framework and shows some evaluation metrics that consider the time of classification. Section 4 reports experimental results that indicate the effectiveness of the proposal. Section 5 presents conclusions and discusses future work directions.

2 Related Work

As far as we know, the whole ETC problem was initially approached in [3]; although the focus was not on making predictions earlier but on improving the classification performance with a sequential reading approach. There, a Markov decision process (MDP) is proposed with two possible actions: *read* (the next sentence), or *classify*. A classifier is trained to learn good/bad state-action pairs

on a high-dimensional space. A potential drawback of this approach would be the well-known scalability problems of MDPs.

In [4] an adaptation of Naïve Bayes is proposed to tackle the problem of classification with partial information. Although a similar performance to models that read the entire document is achieved, the DMC problem is not addressed. Our work starts from this limitation and tries to solve this issue.

Recently, in [6] the CPI and DMC aspects are both addressed by learning the CPI component and using a simple heuristic rule for DMC that consists in classifying a text as positive when exceeding a specific confidence threshold in the prediction of the classifier. The problem with that DMC approach is that is very dependent on the problem and put all the burden of selecting the appropriate thresholds on the ETC system's implementer.

Nevertheless, if we consider the problem of early classification in general (not restricting it to text), we can find different groups that have tackled the problem. For example, in [2] the problem of early classification of time series is formalized as a sequential decision problem involving two costs: quality and delay of the prediction. The method also provides the estimated time for classification, that is, how much of the remaining time series is needed to classify.

3 Early Text Classification Framework

Early text classification focus on the development of predictive models that determine the category of a document as soon as possible. It is assumed that the documents are read sequentially, starting at the beginning of the document and reading words in the order they appear. The objective is to predict the class of a document with as little information (as soon) as possible. In an abstract way, it is like the classic text categorization problem, that is, for a given document it is classified under the class that best fit what had been seen in the training phase. However, differences appear when we want to measure performance. While for classic text categorization problems we use measures like accuracy, precision, recall and the F_1 measure, for early text classification those are not enough. We need measurements that consider the *time* of (*delay* in) the prediction.

This need for temporal evaluation remarks two problems that are related and complementary to each other: classification with partial information and decision of the moment of classification. Our framework defines the way the initial corpus should be divided to train and test this task. The initial corpus is divided into a training and test sets, the first one to train the early text classifier and the second one to test it. We will denote the training set as Tr and the test set as Te. What follows describes the construction of the corpus and classifiers for the different parts of the problem.

3.1 Classification with Partial Information

The task of classification with partial information consists in obtaining an effective predictive model that predicts the class of a document when only partial

information sequentially read up to a certain point of time is available. To achieve this, it is necessary to evaluate the performance of the model on partial documents. It should be noted, however, that when it comes to training the model, the entire document is used.

Given the training set Tr, we partition it in a training set for CPI, denoted TrP, and a test set for CPI, denoted TeP. Since we want to evaluate the performance on partial documents, we must modify the test set for CPI. To achieve this, it is necessary to define a window value $w \in \mathbb{N}$, which indicates the number of terms that are read in each time step. In this way, if $d_j = (t_1, \cdots, t_r)$ is the j-th document in TeP with r the number of terms in the document d_j and $t_1, \cdots, t_r$ the terms in the order they appear in it, then the documents $d_{j,1} = (t_1, \cdots, t_w)$, $d_{j,2} = (t_1, \cdots, t_{2w})$, $\cdots$ and $d_{j,\frac{r}{w}} = (t_1, \cdots, t_{\frac{r}{w}w})$ are part of the pumped test set TeP$'$. This process is repeated for all the document in TeP. For example, given the document "Do not look a gift horse in the mouth" from the test set TeP and $w = 3$, then "Do not look", "Do not look a gift horse" and "Do not look a gift horse in the mouth" will belong to TeP$'$.

Once the training and tests sets were constructed, a model was obtained with machine learning techniques. Nonetheless, the evaluation method was adapted to consider the performance of the model as the reading of the partial documents proceed. For this purpose, subsets of TeP$'$ with the same number of documents as the initial TeP test set were generated. Subset TeP$'_t$ is defined as all partial documents of TeP where the length (in number of terms) is less than or equal to $w \cdot t$. Also, if there exists $d_{j,l}$ and $d_{j,l+w}$ partial documents of d_j with l multiple of w and $l < l + w \leq w \cdot t$, only the largest partial document will belong to TeP$_t$, in this case $d_{j,l+w}$. Thus, the model performance is calculated for the different TeP$_t$ subsets by evaluating it as the terms of the windows are read.

3.2 Context Information

Trying to decide when to stop reading a document only using the class the CPI model returns is difficult. For this reason, we augment the data the DMC model gets with context information, that is, data from the body of the document that could be helpful for deciding the moment of classification. We propose to obtain these from three different sources:

- *Current document*: characteristics relative to the content of the current document. For example, number of: terms, different terms, most relevant terms for each class, stop words, etcetera.
- *Output from CPI*: features produced by the CPI model. This can be the class predicted, current window number and additional information from the model (this depends on the type of model picked for CPI). Regarding the latter, in the case of probabilistic classifiers we can have the probability assigned to each class.
- *Historic data*: related to the context information obtained in previous windows. That is, we apply an aggregation function δ (average, max, min, count, or other function) to some context information from previous window. An

example of historic information is the average probability given to all classes by the CPI model.

The features that in the end are provided to the DMC model depend on the problem under consideration and which are estimated to be more informative to this decision problem. The context information is calculated only for the test set TeP$'$ since the training and test sets for DMC are constructed from it.

3.3 Decision of the Moment of Classification

The task of decision of the moment of classification is to determine the point at which the reading of the document can be stopped with some certainty that the prediction of the classification made is correct. Given a feature vector, this model predicts if we should stop or keep reading. It is expected that when the model for DMC decides to stop reading, the class predicted by the model for CPI is the right one.

The initial corpus for DMC is constructed based in the context information, that is, the training and test sets are formed by the feature vectors generated based in the partial documents of TeP$'$. The problem here is that we do not have the labels indicating when to stop reading. They must be manually obtained or an automatic process should be devised to do it. Here we propose an automatic way to label the feature vector: if the category chosen by CPI is the correct one then the reading can be stopped; if, however, this is not correct we should keep reading. Finally, the corpus is divided into a training set TrM and a test set TeM. The construction and evaluation of the DMC model does not present any particularity, reason why any machine learning technique could be applied.

3.4 Architecture

Once the CPI and DMC models and the context information procedure have been defined, we can formalize the final architecture for our ETC framework. Figure 1 shows the role every model fulfills: CPI is responsible for predicting the category of the partial document, the context information procedure builds the feature vector and, finally, DMC is the one in charge of making the decision of the moment in which the reading of the document must be stopped.

A document classified with this architecture is processed in seven steps:

1. Read w contiguous terms;
2. Build the vector representation of the partial document for CPI;
3. Classify the partial document with CPI;
4. Build the feature vector for DMC;
5. Classify using DMC;
6. If DMC suggests keeping reading terms, return to point 1;
7. If DMC suggests stopping reading then return the category chosen by CPI for the partial document.

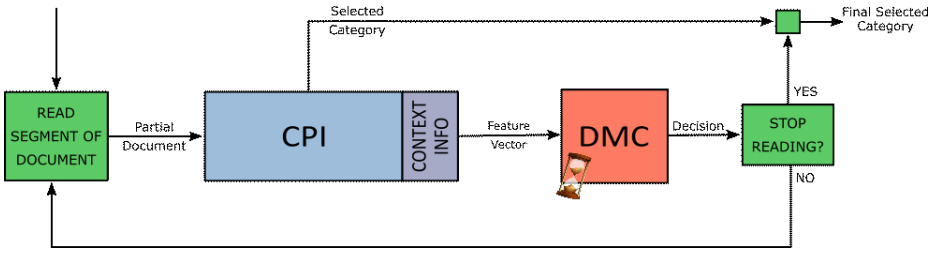

Fig. 1. Early text classification architecture.

3.5 Evaluation Metric

Since the ETC problem has not been addressed for many of the possible configurations, many aspects of it have not been yet defined. Among these is the evaluation of the model, since there is no measure to evaluate the temporary performance of it in a multi-class context. There exists, nonetheless, an evaluation metric for binary early classification [6] that considers the accuracy of the prediction and the delay taken by the system to make the decision. Here the delay is measured by counting the number of terms seen before giving the answer. Given a decision d made by the model at time k, the early risk detection error (ERDE) is defined as:

$$\mathrm{ERDE}_o(d,k) = \begin{cases} c_{\mathrm{fp}} & \text{if the decision } d \text{ is incorrectly positive} \\ c_{\mathrm{fn}} & \text{if the decision } d \text{ is incorrectly negative} \\ lc_o(k) \cdot c_{\mathrm{tp}} & \text{if the decision } d \text{ is correctly positive} \\ 0 & \text{if the decision } d \text{ is correctly negative} \end{cases} \tag{1}$$

The values given to c_{fp} and c_{fn} depends on the application domain and the implications of false positives and false negatives decisions. The factor $lc_o(k) \in [0,1]$ encodes a cost associated to the delay in detecting true positives. In domains where late detection has severe consequences we should set c_{tp} to c_{fn}, that is, late detection is equivalent to not detecting the case at all. The function $lc_o(k)$ should be a monotonically increasing function of k. Losada and Crestani suggest:

$$lc_o(k) = 1 - \frac{1}{1 + e^{k-o}} \tag{2}$$

This function is parametrized by o, which controls the point where the cost grows more quickly. The overall error will be the mean of the ERDE values for all the documents.

Based on this metric, we propose a generalization for ETC when there are more than two classes. ERDE is redefined for each class i as:

$$\mathrm{ERDE}_o(d,i,k) = \begin{cases} c^i_{\mathrm{fp}} & \text{if the decision } d_i \text{ is incorrectly positive} \\ c^i_{\mathrm{fn}} & \text{if the decision } d_i \text{ is incorrectly negative} \\ lc_o(k) \cdot c^i_{\mathrm{tp}} & \text{if the decision } d_i \text{ is correctly positive} \\ 0 & \text{if the decision } d_i \text{ is correctly negative} \end{cases} \tag{3}$$

where d represents the decision made for all the categories, i the category on which the error is being calculated, d_i the decision on category i and k the time when the decision is made. Constants c^i_{fp}, c^i_{fn} and c^i_{tp} indicate the cost associated with the decision on the category being false positive, false negative or true positive, respectively. The values given to these constants depend on the particular addressed problem. Function $lc_o(k)$ is defined as before.

A limitation of Equation (2) is that it does not consider the length of the document. It does not make sense to have one fixed point of penalization when documents have very dissimilar lengths. For instance, if the corpus contains papers and books, a fixed penalization point will harm one of them depending if the document is short or long. We proposed an alternative function to tackle this problem:

$$lc_o(k) = \frac{k}{o} \tag{4}$$

where o represents the length of the document measured in number of terms and k represents the number of terms read at the time of stopping the reading.

Then the early detection error (EDE) for a document is given by the sum of the ERDE for all categories. That is:

$$\text{EDE}_o(d, k) = \sum_{i=1}^{|\mathcal{C}|} \text{ERDE}_o(d, i, k) \tag{5}$$

Since only one category can be chosen by the model (single label problem) and the cost associated with true negatives is zero then the early detection error is reduced to:

$$\text{EDE}_o(d, k) = \begin{cases} lc_o(k) \cdot c^i_{\text{tp}} & \text{if the decision } d_i \text{ is correctly positive} \\ c^j_{\text{fn}} + c^i_{\text{fp}} & \text{if the decision } d_j \text{ is incorrectly negative} \\ & \text{and if the decision } d_i \text{ is incorrectly positive} \end{cases} \tag{6}$$

where the category i is chosen by the CPI and, in the case of misclassification, j is the correct category.

Overall early detection error is obtained by averaging on all documents.

4 Experiments and Results

To test the feasibility of our approach, we used in the experiments the well-known dataset R8 [1]. Based on the Reuters-21578 collection, R8 contains documents belonging to the eight classes with the highest number of training documents in that collection. These documents belong to only one class, thus allowing the corpus to be used for single-label text categorization. More detailed information of R8 is shown in Table 1.

For the experiments, the corpus was processed as follows: a bag-of-words representation of the documents was obtained using the TMG toolbox with a term-frequency weighting scheme [7]. Then, we split the corpus in training and

Table 1. Composition of the corpus R8.

Class	# train doc	# test doc	total # doc
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326
Total	**5485**	**2189**	**7674**

test set for the early text classification model in general and CPI. A window size $w = 3$ was chosen, that is, three terms were read between each run of the early text classification framework.

Based on results obtained in [4] we trained a Naïve Bayes classifier for the CPI model. The performance for the partial documents can be seen in Fig. 2. Clearly, we can accurately classify documents without reading all terms.

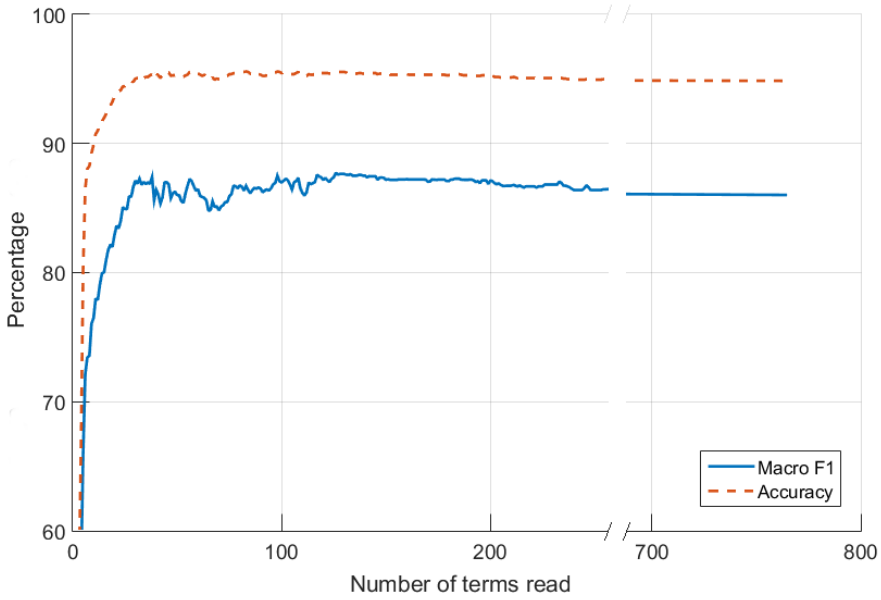

Fig. 2. Evaluation of the sets TeP_t.

Next, we needed to find out what features we could extract to decide *when* to classify a document. In the present work, the selected features were:

– Number of terms of the partial document.
– Number of distinct terms of the partial document.
– Number of relevant terms for each class. Here, we used the most frequent terms in each class.
– Class score given by the CPI model.
– Current window number.
– Historic data from previous windows. For each class we calculated the mean class score given by the CPI model in previous windows.

Finally, we used these features from the documents to build the training and test sets for the DMC model. We tested models trained with three different approaches provided by MatLab: *Naïve Bayes*, *k-ridge*, and an ensemble method (*GentleBoost*) which used neural networks as week classifiers. The performance of each classifier is shown in Table 2. From these results, we chose the GentleBoost classifier for the DMC model.

Table 2. Final estimation results for the precision, recall and F_1 measure.

Predictive DMC Model	**Precision Estimation**	**Recall Estimation**	F_1 **Measure**
Kridge	54.46%	57.52%	55.94
Naïve Bayes	56.19%	56.79%	56.48
Gentleboost	**60.12%**	**78.87%**	**68.23**

Once the CPI and DMC models were trained, our framework was tested and results are shown in Table 3. There, EDE^1_{10} is the early detection error using the definition of $lc_o(k)$ that does not consider the length of the document with $o = 10$, and EDE^2 is the one that considers the length of the document. It can be noted that, for the F_1 measure (measure not considering time), a standard (full reading) model obtains a better result than the temporal one, although the result of our approach is still acceptable. However, when evaluating the temporal aspects, the advantages of the proposal presented in this work are evident: with respect to EDE^1_{10}, a reduction of 0.73 to 0.57 is achieved while for the EDE^2 error the reduction is of 1.05 to 0.73. It can also be observed that there is an average saving of 41.21 terms from the temporal versus the standard approach. Considering the average size of the documents in the R8 collection is 150 terms, this is a significant number of terms that are saved (28% of terms in average).

Table 3. Final results of the linear against the temporal model.

Type of Model	F_1 **Measure**	**Average Unread Terms**	EDE^1_{10}	EDE^2
Standard	85.97	0	0.73	1.05
Temporal (ETC architecture)	78.99	41.21	0.57	0.73

5 Conclusions

Early text classification has not been yet studied in depth, but numerous new applications in on-line detection and real-time systems give a new impetus to research works in this area. This trend has been evident in 2017 where the first conference directly related to ETC was organized and a new one is planned for 2018 (http://early.irlab.org/).

In this paper, we have formalized a simple framework that makes explicit two critical parts in ETC systems: (i) the classification with partial information, and (ii) the decision of the moment of classification. In this context, a novel approach that learns the second component is proposed and a new measure for evaluating multi-class ETC problems is defined.

While promising results were obtained with the R8 corpus reducing considerably the number of terms needed for classification, several directions for future improvements are easily identified. First of all, we can boost up the basic implementation used in the present work by augmenting the contextual information of the DMC model with other more informative features (for instance, using the words with the highest information gain as relevant words). Furthermore, different document representations and predictive models should be tested for CPI and DMC. Finally, we should test this framework in a more recent and competitive early classification corpus like the one presented by Losada and Crestiani [6] and also on other data sets where ETC approaches can be critical like the detection of sexual predators in chats or detection of suicidal discourse.

References

1. A. Cardoso-Cachopo. *Improving Methods for Single-label Text Categorization*. PhD thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.
2. A. Dachraoui, A. Bondu, and A. Cornuéjols. Early classification of time series as a non myopic sequential decision making problem. In *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2015*, volume 9284 of *Lecture Notes in Computer Science*, pages 411–423, Cham, 2015. Springer.
3. G. Dulac-Arnold, L. Denoyer, and P. Gallinari. Text classification: A sequential reading approach. In *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, pages 411–423, Berlin, Heidelberg, 2011. Springer.
4. H.J. Escalante, Montes-y-Gómez M., L.V. Pineda, and M.L. Errecalde. Early text classification: a naïve solution. In *Proc. of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@NAACL-HLT 2016*, pages 91–99, San Diego, California, USA, 2016.
5. A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of the 25th Int. Conf. on NIPS (NIPS'12) - Volume 1*, pages 1097–1105, USA, 2012. Curran Associates Inc.
6. D.E. Losada and F. Crestani. A test collection for research on depression and language use. In *Proc. of Conference and Labs of the Evaluation Forum (CLEF 2016)*, pages 28–39, Evora, Portugal, 2016.
7. D. Zeimpekis and E. Gallopoulos. *Grouping Multidimensional Data: Recent Advances in Clustering*, chapter TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections. Springer, Berlin, Heidelberg, 2006.