Textual-Content-Based Classification of Bundles of Untranscribed Manuscript Images

Jose Ramón Prieto, Vicente Bosch, Enrique Vidal
PRHLT Research Center
Universitat Politècnica de València, Spain
email: joprfon@prhlt.upv.es

Carlos Alonso, M. Carmen Orcero, Lourdes Marquez
Centro de Arqueología Sunacuática
Insituto Andaluz del Patrimonio Histórico, Sevilla, Spain
email: carlos.alonso.v@juntadeandalucia.es

***Abstract*—Content-based classification of manuscripts is an important task that is generally performed in archives and libraries by experts with a wealth of knowledge on the manuscript's contents. Unfortunately, many manuscript collections are so vast that it is not feasible to rely solely on experts to perform this task. Current approaches for textual-content-based manuscript classification generally require the handwritten images to be first transcribed into text – but achieving sufficiently accurate transcripts are generally unfeasible for large sets of historical manuscripts. We propose a new approach to perform automatically this classification task which does not rely on any explicit image transcripts. It is based on "probabilistic indexing", a relatively novel technology which allows to effectively represent the intrinsic word-level uncertainty generally exhibited by handwritten text images. We assess the performance of this approach on a large collection of complex manuscripts from the *Spanish Archivo General de Indias*, with promising results. To the best of our knowledge, this is the first published work proposing, developing and assessing a successful approach for content-based classification of untranscribed manuscript images.**

I. Introduction

We consider the task of automatic classification of bundles or folders (hereafter called "documents") of manuscripts, according to their textual contents. This task has countless applications in libraries and archives where billions of manuscripts are stored without a sufficiently useful identification of their contents.

We assume that the manuscripts of interest have been scanned into high-resolution digital images and the task consists in classifying a given document, that may range from a few to a few thousands of handwritten text images, into a predefined set of classes. Classes are associated with the topic or (semantic) content conveyed by the text written in the document images. When we say "classification of handwritten text images by their textual contents" it is advisable to avoid some frequent confusions.

First of all, this task is very different from what in the computer vision and image analysis literature is usually called "image classification" [1], [2], [3], where images are classified according to more or less global features related to colours, textures, shapes, etc. It is also very different from the task often referred to as "content-based image classification" [4], [5]. In a conventional content-based image classification task, images typically contain a few relatively large objects, such as mountains, animals, vehicles, persons and so forth, out of a few tens (or maybe a few thousands) types of objects. In contrast, a typical text image contains several hundreds of small and detailed "objects" (i.e., words), out of several tens (or hundreds) of thousands "types" of different "objects" (i.e., different words of a natural language lexicon). For similar reasons, works such as [6], [7], where visual and text features are combined, are not comparable with the work here presented.

Another mix up which is worth avoiding is to relate the task considered here with what in the document analysis literature is often called "image document classification", where images of printed or handwritten text are classified according to more or less global features such as layout visual shape, type of script, writer (hand), etc. [8], [9], [10].

Instead, what we intend to do is similar to the time-honoured and well known task of *content-based document classification*, which assumes the data are plain text documents, rather than handwritten text image documents. Traditional examples, for which popular datasets are available, are *Twenty News Groups*, *Reuters*, *WebKB*, etc. [11], [12], [13].

For the task here considered (textual-content-based handwritten text image document classification), the current commonly accepted wisdom is to split the process into two sequential stages. First a handwritten text recognition (HTR) system should be used to transcribe the images into text and, second, traditional content-based document classification methods can be applied to the resulting text documents.

This approach might work to some extent for simple manuscripts with uniform writing style and good quality images, where HTR can provide highly precise transcripts with over 90% word recognition accuracy [14]. It, of course, can also work for small-scale collections, where manual correction of HTR errors can be affordable.

But this is not an option for countless large historical collections of, say, hundreds of thousands of images. Moreover, for many of these collections, the best available HTR systems can only provide word recognition accuracies as low as 50-70% (e.g the ICDAR-2015 benchmark [14]). This is the case of the CARABELA collection considered in this paper. It encompasses more than the 125 000 complex page images [15] and the average word recognition accuracy achieved in optimistic laboratory conditions is 65% [16], dropping to 46% when conditions are closer to real-world usage [15].

Clearly, for this kind of manuscript collections, the aforementioned two-stage idea is to be ruled out and new, holistic approaches should be devised. To the best of our knowledge, this is the first paper proposing, developing and assessing this kind of approaches on a large manuscript dataset – notwithstanding previous publications dealing with related problems and ideas, mainly aimed at printed text [17], [18], [19].

The approach here proposed strongly relies on the so-called *probabilistic indexing* technology, recently developed to deal with the intrinsic word-level *uncertainty* generally exhibited by handwritten text and, more so, by historical handwritten text images [20], [21], [22], [23], [24]. The probabilistic index of a text image can be seen as a "heat-map" image representation which highlights positions of words and "pseudo-word" character sequences which were likely written in the original manuscript. This technology was primarily developed to allow search and retrieval of textual information in untranscribed manuscript collections. In fact, it has recently been successfully applied to allow textual searching in several large collections of untranscribed manuscripts[1] [21], [25], [15].

But probabilistic indexing can go far beyond search and retrieval applications: Since a probabilistic index of a text image provides a distribution of likely words, it allows to properly estimate statistical expectations of the text features required by most plain text content-based document classification methods.

This is the main idea proposed in this paper, which will be developed in Sec. III–IV. Prior to this, the required background technologies will be outlined in Sec. II. The CARABELA project will be briefly presented in Sec. V, along with empirical setting details, and experiments and results will be presented in Sec. VI. The paper will finish in Sec. VII with conclusions and prospects for future work.

II. BACKGROUND

A. Plain Text Document Classification

If a text document is given in some electronic form, its words can be trivially identified as discrete, unique elements, and then the whole field of *text analytics* [11], [13] is available to approach many document processing problems, including *document classification* (DC).

In the traditional field of DC, the text of a document is formalized as a *sequence of words*, where each word is an element of a generally large set called *vocabulary* or *lexicon*. We will use V to denote a vocabulary and v to denote an element of V (i.e., a word). Finally, we will use w to denote a sequence of words; i.e., a *text*.

The most popular document representation model for DC is known as the *bag of words* (BOW) or, more generally, the *vector model* [26], [11], [13]. In this model, the order of words in the text is ignored and a document is represented as a *feature vector* indexed by V. Let $\mathcal{D}$ be a set of documents, $D \in \mathcal{D}$ a document, and $\vec{D} \in \mathbb{R}^N$ the BOW representation of D, where $N \stackrel{\text{def}}{=} |V|$. For each word $v \in V$, $D_v \in \mathbb{R}$ is the value of the v-th feature of $\vec{D}$, which is generally related to the frequency with which v appears in D.

Each document is assumed to belong to a unique class c out of a finite number C of classes. The aim is to predict the best class for any given document, D. Under the vector model, many pattern recognition and machine learning approaches are available. Traditional methods are the so-called Multinomial Naive Bayes (MNB), as well as the (plain) Perceptron and the Support Vector Machines (SVM), among others [11], [12], [13]. Recently, more sophisticated models have been proposed and used, such as Multi-Layer Perceptron (MLP), using or not word embeddings, and even recurrent neural networks such as BLSTMs, which explicitly take into account word order [27], [28], [29], [30]. In this work, we will present results using MNB, and various configurations of MLPs.

1) Feature Selection: Not all the words used in $\mathcal{D}$ are equally helpful to predict the class of a document D. Therefore, a classical first step in DC is to determine a "good" vocabulary, V_n, of reasonable size $n < N$. One of the best ways to determine such a vocabulary is to compute the *information gain* (IG) of each word appearing in $\mathcal{D}$ and retain in V_n only the n words with highest IG.

Loosely following the notation used in [31], let t_v be the value of a boolean random variable which is *True* if, for some random D, v appears in D and *False* otherwise. So, $P(t_v)$ is understood as the probability that some document contains the word v and $P(\bar{t}_v) = 1 - P(t_v)$ as the probability that *no* document contains v. The IG of a word v is then defined as:

$$\begin{aligned} \mathrm{IG}(v) = & - \sum_{c \in C} P(c) \log P(c) \\ & + P(t_v) \sum_{c \in C} P(c \mid t_v) \log p(c \mid t_v) \\ & + P(\bar{t}_v) \sum_{c \in C} P(c \mid \bar{t}_v) \log P(c \mid \bar{t}_v) \end{aligned} \quad (1)$$

where $P(c)$ is de prior probability of class c, $P(c \mid t_v)$ is the conditional probability that a document belongs to class c, given that it contains the word v, and $P(c \mid \bar{t}_v)$ is the conditional probability that a document belongs to class c, given that it does *not* contain v. Note that the first addend of Eq. (1) does not depend on v and can be ignored to rank all $v \in V$ in decreasing order of $\mathrm{IG}(v)$.

To estimate the relevant probabilities in Eq. 1, let $f(t_v) \leq M \stackrel{\text{def}}{=} |\mathcal{D}|$ be the number of documents in $\mathcal{D}$ which contain v and $f(\bar{t}_v) = M - f(t_v)$ the number of those which do *not* contain v. Let $M_c \leq M$ be the number of documents of class c, $f(c, t_v)$ the number of these documents which contain v and $f(c, \bar{t}_v) = M_c - f(c, t_v)$ the number of those that do *not* contain v. Then, the relevant probabilities used in Eq. (1) can be estimated as follows:

$$P(t_v) = \frac{f(t_v)}{M} \qquad P(\bar{t}_v) = \frac{M - f(t_v)}{M} \quad (2)$$

$$P(c \mid t_v) = \frac{f(c, t_v)}{f(t_v)} \qquad P(c \mid \bar{t}_v) = \frac{M_c - f(c, t_v)}{M - f(t_v)} \quad (3)$$

[1] See http://transcriptorium.eu/demots/KWSdemos/ for a list of public search interfaces for these collections.

2) Computing Feature Values: Using information gain, the words $v \in V$ can be sorted by decreasing order of IG(v). Then an adequate vocabulary size, n, can be empirically tuned by using the n first elements of the sorted list to develop a classifier with adequately good precision.

For a given document D, the input to the classifier is a feature vector $\vec{D} \in \mathbb{R}^n$. As previously commented, the value D_v of each feature v is typically related with the frequency of v in D. Let $f(v, D)$ be this frequency – i.e., the number of times v appears in D. One could just define $D_v = f(v, D)$. However, absolute word frequencies can dramatically vary with the size of the documents. Let $V_n(D)$ be the set of words in V_n which appear in D and let $f(D) = \sum_{v \in V_n} f(v, D)$ be the total (or "running") number of words in D. So, for each $v \in V_n$, its normalized frequency, $f(v, D) \,/\, f(D)$, is generally preferred. This ratio, denoted $\mathrm{Tf}(v, D)$ and often called *term frequency*, is a max-likelihood estimate of the conditional probability of word v, given a document D, $P(v \,|\, D)$.

While Tf adequately deals with document size variability, it has been argued that better DC accuracy can be achieved by further weighting each feature with a weight that reflects its "importance" to predict the class of a document. Of course, IG could be used for this purpose, but the so-called *inverse document frequency* (Idf) [32], [33], [34] is argued to be preferable. Idf is defined as $\log(M \,/\, f(t_v))$, which, according to Eq. (2), can be written as $-\log P(t_v)$.

Putting it all together, to represent a document D by a feature vector $\vec{D}$, the value of each feature, D_v, is computed as the Tf·Idf of D and v; i.e., $\mathrm{Tf}(v, D)$, weighted by $\mathrm{Idf}(t)$:

$$\begin{aligned} D_v = \ & \mathrm{Tf}{\cdot}\mathrm{Idf}(v, D) && = \ \mathrm{Tf}(v, D) \cdot \mathrm{Idf}(v) \\ = \ & P(v \,|\, D) \log \frac{1}{P(t_v)} && = \ \frac{f(v, D)}{f(D)} \log \frac{M}{f(t_v)} \end{aligned} \tag{4}$$

B. Probabilistic Indexing of Handwritten Text Images

The Probabilistic Indexing (PrIx) framework was proposed to deal with the intrinsic word-level uncertainty generally exhibited by handwritten text in images and, in particular, images of historical manuscripts. It draws from ideas and concepts previously developed for keyword spotting, both in speech signals and text images. However, rather than caring for "key" words, any element in an image which is likely enough to be interpreted as a word is detected and stored, along with its *relevance probability* (RP) and its location in the image. These text elements are referred to as *"pseudo-word spots"*.

Keyword spotting can be seen as a binary classification problem to decide whether a particular image region x is *relevant* for a given query word v, i.e. try to answer the following question: "Is v actually written in x?". As in [20], [23], we denote this image-region word RP as $P(R{=}1 \mid X{=}x, V{=}v)$, but for the sake of conciseness, we will omit the random variable names, and for $R = 1$, we will simply write R. As discussed in [35], this RP can be simply approximated as:

$$P(R \mid x, v) = \sum_{b \sqsubseteq x} P(R, b \mid x, v) \approx \max_{b \sqsubseteq x} P(v \mid x, b) \tag{5}$$

where b is a small, word-sized image sub-region or Bounding Box (BB), and with $b \sqsubseteq x$ we mean the set of all BBs contained in x. $P(v \mid x, b)$ is just the posterior probability needed to "recognize" the BB image (x, b). Therefore, assuming the computational complexity entailed by the maximization in (5) is algorithmically managed, any sufficiently accurate isolated word classifier can be used to obtain $P(R \mid x, v)$.

Alternatively, $P(R \mid x, v)$ can be computed using a suitable segmentation-free *word-sequence* recognizer [20], [35], [23]:

$$P(R \mid v, x) \ = \sum_{w} P(R, w \mid v, x) \ = \sum_{w : v \in w} P(w \mid x) \tag{6}$$

where w is the sequence of words of an (unknown) transcript of x and with $v \in w$ we mean that v is one of the words of w. So the RP can be computed using state-of-the-art optical and language models and processing steps similar to those employed in handwritten text recognition, even though no actual text transcripts are explicitly produced in PrIx. In this work we have adopted the PyLaia HTR toolkit [36], which has proved to provide excellent modelling (and recognition) performance in many HTR tasks [23], [14].

Image region word RPs do not take explicitly into account where the considered words may appear in the region x, but the precise positions of the words within x are easily obtained as a by-product.

The PrIx approach usually adopts character-level optical and language models, but it achieves good performance for word queries by determining RPs for *"pseudo-words"*. As previously commented, pseudo-words are arbitrary character sequences that are likely-enough to be actual words and which are automatically "discovered" in the very test images being indexed [37], [23].

This word-level indexing approach has proved to be very robust, and it has been used to very successfully index several large iconic manuscript collections, such as the Medieval French CHANCERY collection [21], the BENTHAM PAPERS [25], and the Spanish CARABELA collection considered in this paper, among others.[2]

III. BASIC TEXT ANALYTICS USING PRIX

The primary usage of the PrIx of a manuscript image collection is to allow fast and accurate search for textual information in the images. However, the information contained in a PrIx can be useful for many other text analytics applications which can be based on incomplete and/or imprecise textual contents of the images. One of these applications, considered in this paper, is image document classification by textual content. In preparation for this application, here we discuss how PrIx can be used to estimate basic features of the text accurately.

Since R is a binary random variable, the RP $P(R \mid x, v)$ can also be properly seen as the statistical expectation that v is written in the region x. Therefore, the sum of RPs for all the pseudo-words indexed in x should approach the number of words written in x, $n(x)$. Formally speaking, let w be a word

[2] See: http://transcriptorium.eu/demots/KWSdemos

sequence corresponding to the (unknown) transcript of x and let $n(w) = |w|$. Then, the expected value of $n(x)$ is [25]:

$$E[n(x)] = \sum_{v} P(R \mid x, v) \tag{7}$$

In practice, if target image regions are sufficiently small (e.g., short lines), most of the words in a region are typically different and this is generally a good, lower bound approximation to the total number of words written in x.

Eq. (7) can easily be extended to estimate the total number of *running words* of a page image or a larger document, X, containing several pages (i.e., $f(D)$, see II-A2), or even the full dataset. If $x \sqsubseteq X$ is understood as the set of all the indexed regions (e.g. line images) of X, the expected number of words in X, $n(X)$ is:

$$E[n(X)] = \sum_{x \sqsubseteq X} \sum_{v} P(R \mid x, v) \tag{8}$$

Finally, the frequency of a specific (pseudo-)word v in the transcript of a document X, $n(v, X)$, can be estimated as:

$$E[n(v, X)] = \sum_{x \sqsubseteq X} P(R \mid x, v) \tag{9}$$

The accuracy of some of these estimates for two manuscript collections was empirically studied in [25], [15], where estimation errors well below 10% are reported.

Word occurrence estimates can be useful for many applications, including image document classification as will be discussed below. Another word and document related frequency, useful for document classification, is the number of documents in a collection, $\mathcal{X}$, which contain a given word, v. The probability that v is written in a document X can be approximated by considering this binary event as a boolean OR combination of the events that v is written in each of the image regions x of X. And, as discussed in [24], $P(R \mid X, v) \approx \max_{x \sqsubseteq X} P(R \mid x, v)$. Therefore, the expected number of documents which contain the word v, $m(v, \mathcal{X})$ can be approximated as:

$$E[m(v, \mathcal{X})] = \sum_{X \sqsubseteq \mathcal{X}} \max_{x \in X} P(R \mid x, v) \tag{10}$$

IV. Textual-Content-Based Image Classification

A. Estimating Information Gain and Tf·Idf from PrIx's

Traditional techniques for plain text document classification, based on the BOW or vector representation model, have been outlined in Sec. II-A. All these techniques ultimately relay on frequencies of words per document, or per document and class, and/or frequencies of documents which contain a given word. Obviously, for a collection of untranscribed text images, no text is available to compute these frequencies but, as discussed in Sec. III, they can be *estimated* from the image PrIx's

According to the notation used in Sec: II-B and III, a document D in Sec. II-A becomes a set of text images or *image document*, X. Also, the set of all documents $\mathcal{D}$ becomes the text image collection $\mathcal{X}$ and we will denote $\mathcal{X}_c$ the subset of image documents of class c. Thus $M \stackrel{\text{def}}{=} |\mathcal{X}|$ is now the total number of image documents and $M_c \stackrel{\text{def}}{=} |\mathcal{X}_c|$ the number of them which belong to class c.

The frequencies needed to compute the IG of a word, v are summarized in Eqs. (2–(3). Using the PrIx's of $\mathcal{X}$, the number of documents which contain the word v, $f(t_v) \equiv m(v, \mathcal{X})$, can be directly estimated using Eq. (10). Similarly, the number of documents of class c which contain v, $f(c, t_v)$, can be estimated as in Eq. (10) changing $\mathcal{X}$ with $\mathcal{X}_c$.

On the other hand, the frequencies needed to compute the Tf·Idf document vector features are summarized in Eq. (4). In addition to $f(t_v) \equiv m(v, \mathcal{X})$, already discussed above, we need the total number of running words in a document D, $f(D)$, and the number of times the word v appears in D, $f(v, D)$. Clearly, $f(D) \equiv n(X)$ and $f(v, D) \equiv n(v, X)$, which can be directly estimated using Eq. (8) and Eq. (9), respectively.

B. Image Document Classification

Once we know how to estimate the frequencies needed to compute the word IG and the Tf·Idf vector representation of image documents, optimal prediction of the class of an image document X is achieved under the minimum-error risk statistical framework:

$$c^\star(X) \;=\; \arg\max_{c \in \{1,\ldots,C\}} P(c \mid X) \tag{11}$$

Using a vector representation of X, $\vec{X}$ (e.g., Tf·Idf), the posterior $P(c \mid X)$ can be computed following several well-known approaches. A most direct one, which we mainly consider in this work, is the Perceptron and Multi-Layer Perceptron (MLP).

1) Multilayer Perceptrons: We use architectures where the output is a softmax layer with C units and training is performed by backpropagation using the cross entropy loss. Under these conditions, it is straightforward that each output c for an input vector $\vec{X}$ approaches $P(c \mid X), 1 \leq c \leq C$, and thus Eq. (11) directly applies.

We have considered various architectures with different numbers of layers. In all the cases, every layer except the last one is followed by batch normalization and ReLU activation functions, Three configurations were tested. The basic one, is a plain multiclass perceptron where the input is totally connected to each of the C neurons of the output layer (hence no hidden layers are used). For the sake of simplifying and homogenizing the terminology, here we consider such a model as a "0-hidden-layers MLP" and refer to it as MLP-0. The next configuration, MLP-1, was a proper MLP including one hidden layer with 64 ReLU neurons and batch normalization [38]. The hidden layer was expected to do some kind of intra-document clustering, hopefully improving the classification ability of the last layer. Finally, we also tested a deeper model, MLP-3, with 3 hidden layers including 16, 32 and 64 ReLU neurons and batch normalization.

2) Multinomial Naive Bayes and Plain Perceptron: Other popular approaches use the Bayes' rule to transform Eq. (11) into:

$$c^\star(X) \;=\; \underset{c\in\{1,\ldots,C\}}{\arg\max}\; P(c)\,P(X \mid c) \qquad (12)$$

One of these approaches is the MNB classifier [11], [39], in which $P(X \mid c)$ is computed as:

$$P(X \mid c) \;=\; K \prod_{v\in V_n} (\theta_{cv})^{X_v} \qquad (13)$$

where K is a term that does not depend on c, X_v is the v-th component of the image document feature vector $\vec{X}$, and θ_{cv}, $v \in V_n$, are the parameters of the MNB distribution for class c, $1 \leq c \leq C$. In this case, Eq. (12) can be rewritten as:

$$c^\star(X) \;=\; \underset{c\in\{1,\ldots,C\}}{\arg\max}\; \log P(c) + \sum_{v\in V_n} (\log \theta_{cv})\, X_v \qquad (14)$$

which clearly shows that MNB is a linear classifier equivalent to a (plain) perceptron [39]. Nevertheless, it should be pointed out that the accuracy which can be achieved with these classifiers may differ, not only because of the different training criteria adopted, but also because of differences in how the priors $P(c)$ are handled in each case.

Typically $X_v = n(v, X)$, but better results are often reported if rather than raw word frequencies, their normalized and weighted versions, $\mathrm{Tf}\cdot\mathrm{Idf}$ (see Eq. (4)), are used.

V. Dataset and Experimental Settings

The dataset considered in this work is part of the manuscript collection compiled in the CARABELA project. In this section we outline this project and provide details of the dataset and other settings adopted for the experiments discussed in Sec. VI.

A. The Carabela Project

The full manuscript collection considered in the CARABELA project contain more than 125 000 images of manuscripts of interest to underwater archaeology. The images used in the present work correspond to documents from the Archivo General de Indias (AGI), wich encompass about one forth of the CARABELA collection [15].

The very many and extremely variable writing styles, the heavy use of archaisms and non-standard abbreviations [16], the poor quality of original documents and/or scanned images and the sheer size of the collection, makes CARABELA one on the most challenging sets of historical manuscripts we have ever considered.

A basic aim of CARABELA was to obtain PrIx's for all the images of the collection, in order to make it searchable by textual queries, as in previous similar projects [21], [25].

Another important objective was to demonstrate the feasibility of classifying sets (documents) of untranscribed text images according to their textual contents. This is the focus of the present paper. Given the contents of the CARABELA manuscripts, the main classification task demanded by the archive holders was to classify image documents into classes associated with the level of risk of public exposure of the corresponding images. Simplifying matters, the aim was to automatically detect those folders that may be sensitive for the protection of underwater Spanish heritage facing possible abuses such as shipwreck looting.

All objectives of CARABELA were achieved [15]. 1) More than 125 000 page images were probabilistically indexed with satisfactory evaluation results; 2) an effective search system was implemented for real textual information retrieval[3] which is being very positively appraised by expert users and 3) new machine learning classification methods were developed for accurate textual-content-based classification of documents of untranscribed images, as discussed in this paper.

B. Empirical settings

From the AGI part of the CARABELA collection, 199 documents were manually labelled by the paleographers and historian partners of the project. The sizes of these documents varies from five to more than 2000 page images. Three class labels were defined, LOW, MED. and HIGH, depending on the risk that exposing publicly the documents would entail, as discussed in Sec. V-A. The machine learning task thus consists in training a model to classify each document into one of theese three classes (i.e., $C = 3$). To avoid the impact of outliers and to ensure each documents considered has a minimum of information to reasonably allow distinguishing its class, we discarded documents with less than five page images. This resulted in an experimental dataset of $M = 155$ documents, the details of which are shown in Table I. While the amount of data in this dataset is large (21 889 text images), the actual number of samples to be classified is quite small (155 documents). Therefore, to achieve reasonably reliable results, all the experiments presented in Sec. VI, were carried out following a 15-fold cross-validation protocol.

TABLE I
NUMBER OF DOCUMENTS AND PAGE IMAGES: PER CLASS, PER DOCUMENT & CLASS, AND TOTALS.

	LOW	MED.	HIGH	**Total**
Number of Documents	19	67	69	**155**
Average Images per Document	210	193	72	**21 889**
Min–Max Images per Document	5–475	5–2143	5–1733	**5–2143**

PrIx vocabularies are typically huge because they contain a large amount of pseudo-word hypotheses. However, many of these hypotheses have low relevance probability and most of the low-probability pseudo-words are not real words. Therefore, as a first step, we pruned out the huge PrIx vocabulary, avoiding words with less than three characters, as well as words with estimated document frequency lesser than 1.0. This resulted in a vocabulary of 69 279 (pseudo-)words. Secondly, to retain the most relevant features as discussed in Sec. II-A1, (pseudo-)words were sorted by decreasing values of IG and the first n entries of the sorted list were selected to define a BOW vocabulary V_n. Exponentially increasing values of n from 8 up to 16 384 were considered in the experiments. Finally, a $\mathrm{Tf}\cdot\mathrm{Idf}$ n-dimensional vector was calculated for each document.

[3] http://carabela.prhlt.upv.es/en/demonstrators

For MLP classification, document vectors were scaled by subtracting the mean and dividing by the standard deviation, resulting in zero-mean and unit-variance input vectors. The parameters of each MLP architecture were initialized following [40] and trained according to the cross-entropy loss for 50 epochs using the ADAM optimizer [41], with a learning rate of 0.1, $\beta_1 = 0.5$, $\beta_2 = 0.999$. As mentioned above, the models were trained and tested following a 15-fold cross-validation protocol, ending up with the average accuracy for all folds. In each fold, a validation set of 10% of the training data of this fold was used for MLP training. In order to avoid random initialization effects and obtain more consolidated results, for every value of n the whole cross-validation process was executed 100 times with different initialization seeds, and then average results were calculated.

MNB experiments, on the other hand, were carried out with the SKLEARN toolkit [42], using Tf·Idf input features and following the same 15-fold cross-validation protocol as for MLP models (except for random initializations, wich are pointless here). Smoothing is often reported to allow improved MNB results, but our best results were achieved with no smoothing ($\alpha = 0$) – probably because in our task there are only three classes and the overall amount of training data (taking into account the number of pages per document, see Table I) is fairly large. In our classification task classes are fairly unbalanced (see Table I), which is known to raise training issues for MNB. This is studied in detail in [39], where the so-called "Complement MNB" (C-MNB), is proposed to mitigate the resulting data skew. In our experiments we found C-MNB to provide slightly better overall performance than plain MNB.

VI. Experiments and Results

Empirical work has mainly focused on MLP classifiers, but other approaches, such as MNB, have been also studied.

A. Multilayer Perceptrons

Classification error rates are presented in figure 1 for 12 increasing values of n, the number of (pseudo-)words selected with maximum Information Gain. As previously mentioned, each reported result is the average of 100 15-fold cross-validation executions with different random initialization seeds. As discussed in Sec. IV-B1, three architectures were evaluated, referred to as MLP-0, MLP-1 and MLP-3.

Best results are obtained for the plain perceptron (MLP-0), for a relatively large vocabulary of 2048 words. For this model, accuracy remains good if the vocabulary size is reduced down to 128 words. Further reductions, or using vocabularies larger than 2048 words, lead to sharp accuracy degradations. The other (proper) MLP models achieve almost the same accuracy as MLP-0 for the 4096 words vocabulary, and similarly good accuracies for a wider range of vocabulary sizes. Moreover, the accuracy of both MLP-1 and MLP-3 do not fall significantly when the vocabulary size is increased and degrade more gracefully for smaller sizes.

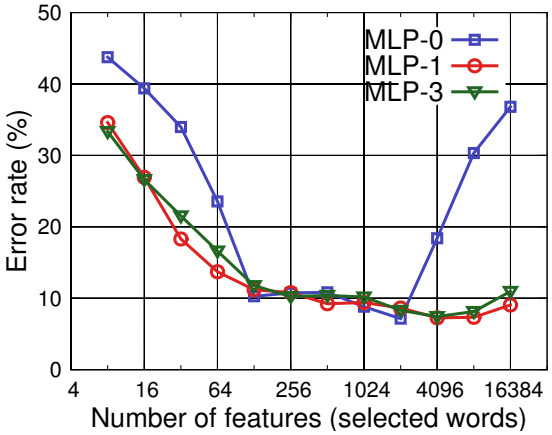

Fig. 1. Classification error rate for three classifiers. MLP-0 is a plain multiclass perceptron, MLP-1 is a multilayer perceptron with a single hidden layer and MLP-3 has 3 hidden layers. Each result is the average of 100 15-fold cross-validation runs with randomly initialized model training. 95% confidence intervals (not shown for clarity) are all smaller than $\pm 1.3\%$.

Model complexity, in terms of numbers of parameters to train, grows with the number of features, n as:

P: $3(n+1)$ MLP-1: $64n + 387$ MLP-3: $16n + 3091$

For all $n > 64$, the least complex model is MLP-0, followed by MLP-3 and MLP-2. For $n = 2048$ MLP-0 has 6147 parameters, while MLP-2 and MLP-3 have 131459 and 35859 parameters, respectively. And for $n = 4096$ the number of parameters of MLP-2 and MLP3 are 262531 and 68627, respectively. Therefore, also taking into account the model complexity, MLP-0 seems to be the best choice for the task considered in this work.

For the best model (MLP-0, with 2048 features), Table II shows the average confusion matrix and the specific error rate per class. It is worth noting that the best accuracy is achieved for the most sensitive (HIGH) class, were false negatives may have the most negative impact on the intended application.

TABLE II
AVERAGE CONFUSION MATRIX FOR 100 15-FOLD CROSS-VALIDATION RANDOMLY INITIALIZED RUNS, USING 2048 WORDS WITH THE GREATEST IG AND THE MLP-0 CLASSIFIER.

	LOW	MED.	HIGH	Total	**Error** (%)
Low	15.7	0.2	3.1	19	**17.1**
Med.	1.0	62.8	3.2	67	**6.2**
High	1.3	2.3	65.4	69	**5.3**
All classes	18.0	65.3	71.7	155	**7.1**

B. Multinomial Naive Bayes

C-MNB classification error rates for increasing numbers of features (n), are reported in Table III. MLP-0 (Perceptron) results for the same values of n (already posted in Fig. 1), are also included to allow direct comparison. The overall superiority of MLP-0 can be clearly observed – and the same could be said for MLP-1 and MLP-3. Since the MLP-0 architecture is just that of a plain multiclass perceptron, it is functionally equivalent to MNB, as discussed in Sec. IV-B2. Therefore the better performance achieved by MLP-0 should be attributed to the different training approach: max posterior (MAP) for MLP-0, versus max likelihood (MLE) for MNB.

TABLE III
CLASSIFICATION RESULTS USING C-MNB, COMPARED WITH MLP-0.

Error (%)	16	64	256	1024	2048	4096	9192	16384
C-MNB	42.6	44.5	26.5	20.0	16.1	14.8	15.5	20.6
MLP-0	39.4	23.6	10.7	8.8	7.1	18.4	30.3	36.8

Additional experiments were carried out with Support Vector Machines. However, the best performance we could achieve was significantly worse than that of C-MNB. These results are not reported for the sake of brevity.

VII. CONCLUSION

We have presented and showcased an approach that is able to perform textual-content-based document classification directly on documents of untranscribed handwritten text images. Our method uses traditional techniques for plaintext document classification, estimating the required word frequencies from image probabilistic indexes. This way, we overcome the need to explicitly transcribe manuscripts, which is generally unfeasible for large collections.

The experimental results obtained with the proposed approach leave no doubt regarding its capabilities to model the textual contents of the page images and to discriminate among content-defined classes.

In our opinion, probabilistic indexing opens new avenues for research in textual-content-based image document classification. In future works, we plan to explore the use of other classification methods based on information extracted from probabilistic indexes. On the other hand, we aim to capitalize on the observation that fairly accurate classification can be achieved with relatively small vocabularies, down to 64 words in the task considered in this paper. In this direction, we will explore the use of information gain and/or $\mathrm{Tf}{\cdot}\mathrm{Idf}$ values estimated for probabilistic index (pseudo-)words to derive a small set of words that semantically describes the contents of each bundle of manuscripts. This would allow the automatic or semi-automatic creation of bundle metadata which could be extremely useful for scholars and general public searching for historical information in archived manuscripts.

ACKNOWLEDGMENTS

Work partially supported by the BBVA Foundation through the 2017–2018 and 2018–2019 Digital Humanities research grants "Carabela" and "HisClima – Dos Siglos de Datos Climáticos". The first author's work was partially supported by the Universitat Politècnica de València under grant FPI-I/SP20190010. Computing infrastructure was provided by the EU-FEDER Comunitat Valenciana 2014-2020 grant ID-IFEDER/2018/025.

REFERENCES

[1] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European conference on computer vision*. Springer, 2010, pp. 143–156.

[2] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and svm training," in *CVPR 2011*. IEEE, 2011, pp. 1689–1696.

[3] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[4] S. B. Park, J. W. Lee, and S. K. Kim, "Content-based image classification using a neural network," *Pattern Recognition Letters*, vol. 25, no. 3, pp. 287–300, 2004.

[5] S. Kumar, Z. Khan, and A. Jain, "A review of content based image classification using machine learning approach," *International Journal of Advanced Computer Research*, vol. 2, no. 3, p. 55, 2012.

[6] S. Paek, C. L. Sable, V. Hatzivassiloglou, A. Jaimes, B. H. Schiffman, S.-F. Chang, and K. R. McKeown, "Integration of visual and text-based approaches for the content labeling and classification of photographs," in *Acm sigir*, vol. 99. Citeseer, 1999, pp. 15–19.

[7] L. Tian, D. Zheng, and C. Zhu, "Image classification based on the combination of text features and visual features," *International journal of intelligent systems*, vol. 28, no. 3, pp. 242–256, 2013.

[8] N. Chen and D. Blostein, "A survey of document image classification: problem statement, classifier architecture and performance evaluation," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 10, no. 1, pp. 1–16, 2007.

[9] L. Kang, J. Kumar, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for document image classification," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 3168–3172.

[10] Y.-J. Xiong, Y. Lu, and P. S. Wang, "Off-line text-independent writer recognition: A survey," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 31, no. 05, p. 1756008, 2017.

[11] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[12] A. Khan, B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," *Journal of advances in information technology*, vol. 1, no. 1, pp. 4–20, 2010.

[13] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.

[14] J. A. Sánchez, V. Romero, A. H. Toselli, M. Villegas, and E. Vidal, "A set of benchmarks for handwritten text recognition on historical documents," *Pattern Recognition*, vol. 94, pp. 122–134, 2019.

[15] E. Vidal et al., "The carabela project and manuscript collection: Large-scale probabilistic indexing and content-based classification," in *16th ICFHR*, Sep 2020.

[16] V. Romero, A. H. Toselli, E. Vidal, J. A. Sánchez, C. Alonso, and L. Marqués, "Modern vs diplomatic transcripts for historical handwritten text recognition," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 103–114.

[17] M. Rusinol, V. Frinken, D. Karatzas, A. D. Bagdanov, and J. Lladós, "Multimodal page classification in administrative document image streams," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 4, pp. 331–341, 2014.

[18] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. Lee Giles, "Learning to extract semantic structure from documents using multimodal fully convolutional neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 5315–5324.

[19] R. Jain and C. Wigington, "Multimodal document image classification," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 71–77.

[20] A. H. Toselli, E. Vidal, V. Romero, and V. Frinken, "HMM Word Graph based Keyword Spotting in Handwritten Document Images," *Information Sciences*, vol. 370-371, pp. 497–518, 2016.

[21] T. Bluche, S. Hamel, C. Kermorvant, J. Puigcerver, D. Stutzmann, A. H. Toselli, and E. Vidal, "Preparatory KWS Experiments for Large-Scale Indexing of a Vast Medieval Manuscript Collection in the HIMANIS Project," in *2017 14th IAPR Int. Conf. on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 311–316.

[22] E. Lang, J. Puigcerver, A. H. Toselli, and E. Vidal, "Probabilistic indexing and search for information extraction on handwritten german parish records," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Aug 2018, pp. 44–49.

[23] J. Puigcerver, "A probabilistic formulation of keyword spotting," Ph.D. dissertation, Univ. Politècnica de València, 2018.

[24] A. H. Toselli, E. Vidal, J. Puigcerver, and E. Noya-García, "Probabilistic multi-word spotting in handwritten text images," *Pattern Analysis and Applications*, vol. 22, no. 1, pp. 23–32, 2019.

[25] A. Toselli, V. Romero, E. Vidal, and J. Sánchez, "Making two vast historical manuscript collections searchable and extracting meaningful textual features through large-scale probabilistic indexing," in *2019 15th IAPR Int. Conf. on Document Analysis and Recognition (ICDAR)*, 2019.

[26] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text classification using machine learning techniques." *WSEAS transactions on computers*, vol. 4,8, pp. 966–974, 2005.

[27] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[28] L. Lenc and P. Král, "Word embeddings for multi-label document classification," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., Sep. 2017, pp. 431–437.

[29] Q. Liu, H. Huang, Y. Gao, X. Wei, Y. Tian, and L. Liu, "Task-oriented word embedding for text classification," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2023–2032.

[30] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–221, 2020.

[31] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization," *Inf. Proc. & management*, vol. 42, no. 1, pp. 155–165, 2006.

[32] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Proc. & Management*, vol. 24, no. 5, p. 513/523, 1988.

[33] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization." Carnegie-mellon univ pittsburgh pa dept of computer science, Tech. Rep., 1996.

[34] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Inf. Proc. & Management*, vol. 39, no. 1, pp. 45–65, 2003.

[35] E. Vidal, A. H. Toselli, and J. Puigcerver, "A probabilistic framework for lexicon-based keyword spotting in handwritten text images," UPV, Tech. Rep., 2017.

[36] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 1, 2017, pp. 67–72.

[37] A. H. Toselli, J. Puigcerver, and E. Vidal, "Two methods to improve confidence scores for lexicon-free word spotting in handwritten text," in *Proc. 15th ICFHR*, 2016, pp. 349–354.

[38] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.

[39] J. D. Rennie, S. Lawrence, J. Teevan, and D. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," *Proc. of the Twentieth Int. Conference on Machine Learning (ICML-2003),*, no. 2003, 2003.

[40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.

[41] K. Diederik P. and L. Ba, "Adam: A Method for Stochastic Optimization," *AIP Conference Proceedings*, vol. 1631, pp. 58–62, 2014.

[42] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.