Text Categorization for Multi-page Documents: A Hybrid Naive Bayes HMM Approach

Paolo Frasconi
Department of Systems and Computer Science
University of Florence
50139 Firenze, Italy
paolo@dsi.unifi.it

Giovanni Soda
Department of Systems and Computer Science
University of Florence
50139 Firenze, Italy
giovanni@dsi.unifi.it

Alessandro Vullo
Department of Systems and Computer Science
University of Florence
50139 Firenze, Italy
alex@mcculloch.ing.unifi.it

ABSTRACT

Text categorization is typically formulated as a concept learning problem where each instance is a single isolated document. In this paper we are interested in a more general formulation where documents are organized as page sequences, as naturally occurring in digital libraries of scanned books and magazines. We describe a method for classifying pages of sequential OCR text documents into one of several assigned categories and suggest that taking into account contextual information provided by the whole page sequence can significantly improve classification accuracy. The proposed architecture relies on hidden Markov models whose emissions are bag-of-words according to a multinomial word event model, as in the generative portion of the Naive Bayes classifier. Our results on a collection of scanned journals from the Making of America project confirm the importance of using whole page sequences. Empirical evaluation indicates that the error rate (as obtained by running a plain Naive Bayes classifier on isolated page) can be roughly reduced by half if contextual information is incorporated.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*; H.3.7 [**Information Systems**]: Information Storage and Retrieval—*Digital Libraries*; I.7.m [**Computing Methodologies**]: Document and Text Processing

General Terms

Algorithms, Performance

Keywords

Text categorization, Hidden Markov Models, Naive Bayes classifier, Multi-page documents

JCDL'01, June 24-28, 2001, Roanoke, Virginia, USA.

1. INTRODUCTION

Text categorization is the problem of grouping textual documents into different fixed classes or categories. The task is related to the ability of an intelligent system to automatically perform tasks such as personalized e-mail or news filtering, document indexing, metadata extraction. These problems are of great and increasing importance, mainly because of the recent explosive increase of online textual information. Text categorization is generally formulated in the machine learning framework. In this setting, a learning algorithm takes as input a set of labeled examples (where the label indicates which category the example document belongs to) and attempts to infer a function that will map new documents into their categories. Several algorithms have been proposed within this framework, including regression models [29], inductive logic programming [6], probabilistic classifiers [17, 21, 16], decision trees [18], neural networks [22], and more recently support vector machines [12].

Research on text categorization has been mainly focused on non-structured documents. In the typical approach, inherited from information retrieval, each document is represented by a sequence of words, and the sequence itself is normally flattened down to a simplified representation called *bag of words* (BOW). This is like representing each document as a feature-vector, where features are words in the vocabulary and components of the feature-vector are statistics such as word counts in the document. Although such a simplified representation is appropriate for relatively flat documents (such as email and news messages), other types of documents are internally structured and this structure should be exploited in the representation to better inform the learner.

In this paper we are interested in the domain of digital libraries and, in particular, collections of digitized books or magazines, with text extracted by an Optical Character Recognition (OCR) system. One important challenge for digital conversion projects is the management of structural and descriptive metadata. Currently, metadata management involves a large amount of keying work carried out by human operators. Automating the extraction of metadata from digitized documents could greatly improve efficiency and productivity [1]. This automation, however, is not a trivial task and involves recognition of the ordering of text divisions, such as chapters and sub-chapters, the identification of layout elements, such as headlines, footnotes, graphs, and captions, and the linking of articles within a pe-

riodical. Automatic recognition of these elements can be a hard problem, especially without any prior knowledge about the type of elements that are expected to be present within a given document page. Hence, page classification can represent a useful preliminary step to guide the subsequent extraction process. Moreover, extracting metadata related to the semantic contents of document parts (such as chapters or articles) can require the ability of recognizing the topic or the category of these parts. The solution to these problems can be helped by a classifier that assigns a category to each page of the document.

Unlike email or news articles, books and periodicals are *multi-page* documents and the simplest level of structure that can be exploited is the serial order relation defined among single pages. The task we consider is the automatic categorization of each page according to its (semantic) contents[1]. Exploiting the serial order relation among pages within a single document can be expected to improve classification accuracy when compared to a strategy that simply classifies each page separately. This is because the sequence of pages in documents such as books or magazines often follows regularities such as those implied by typographical and editorial conventions. Consider for example the domain of books and suppose categories of interest include `title-page`, `dedication-page`, `preface-page`, `index-page`, `table-of-contents`, `regular-page`, and so on. Even in this very simplified case we can expect constraints about the valid sequences of page categories in a book. For example, `title-page` is very unlikely to follow `index-page` and, similarly, `dedication-page` is more likely to follow `title-page` than `preface-page`. Constraint of this type can be captured and modeled using a stochastic grammar. Thus, information about the category of a given page can be gathered not only by examining the contents of that page, but also by examining the contents of other pages in the sequence. Since contextual information can significantly help to disambiguate between page categories, we expect that classification accuracy will improve if the learner has access to whole sequences instead that single-page documents.

In this paper we combine several algorithmic ideas to solve the problem of text categorization in the domain of multi-page documents. First, we use an algorithm similar to those described in [28] and [20] for inducing a stochastic regular grammar over sequences of page categories. Second, we introduce a hidden Markov model (HMM) that can deal with sequences of BOWs. Each state in the HMM is associated with a unique page category. Emissions are modeled by a multinomial distribution over word events, like in the generative component of the Naive Bayes classifier. The HMM is trained from (partially) labeled page sequences, i.e. state variables are partially observed in the training set. Unobserved states (which is the common setting in most classic applications of HMMs) arise here when document pages are partially unlabeled, like in the framework described in [23] and [13]. Finally, we solve the categorization problem by running the Viterbi algorithm on the trained HMM, yielding a sequence of page categories associated with new (unseen) documents. This is somewhat related to recent applications of HMMs to information extraction [9, 20] but the output labeling in our case is associated with the entire stream of text contained into a page, while in [9, 20] the HMM is used to attach labels to single words of shorter portions of text.

Our approach is validated on a real dataset consisting of 95 issues of the journal *American Missionary*, which is part of the "Making of America" collection [26]. In spite of text noise due to optical recognition, our system achieves about 85% page classification accuracy when training on 10 issues (year 1884) and testing on issues from 1885 to 1893. More importantly, we show that incorporating contextual information significantly reduces classification error, both in the case of completely labeled example documents and when unlabeled documents are included in the training set.

2. BACKGROUND

Let $\boldsymbol{d}$ be a generic multi-page document, and let d_t denote the t-th page within the document. The categorization task consists of learning from examples a function $f : d_t \rightarrow \{c^1, \cdots, c^K\}$ that maps each page d_t into one out of K classes.

2.1 The Naive Bayes classifier

The above task can also be reformulated in probabilistic terms as the estimation of the conditional probability $P(C_t = c^k|d_t)$, C_t being a multinomial class variable with realizations in $\{c^1, \cdots, c^K\}$. In so doing, f can be computed using Bayes' decision rule, i.e. $f(d)$ is the class with higher posterior probability. The Naive Bayes classifier computes this probability as

$$P(C_t = c^k|d_t) \propto P(d_t|C_t = c^k)P(C_t = c^k). \qquad (1)$$

What characterizes the model is the so-called Naive Bayes assumption, prescribing that word events (each occurrence of a given word in the page corresponds to one event) are conditionally independent *given* the page category. As a result, the class conditional probabilities can be factorized as

$$P(d_t|C_t = c^k) = \prod_{i=1}^{|d_t|} P(w_t^i|C_t = c^k) \qquad (2)$$

where $|d_t|$ denotes the length of page d_t and w_t^i is the i-th word in the page. This conditional independence assumption is graphically represented by the Bayesian network[2] shown in Figure 1.

Although the basic assumption is clearly false in the real world, the model works well in practice since classification requires finding a good separation surface, not necessarily a very accurate model of the involved probability distributions. Training consists of estimating model's parameters from a dataset $\mathcal{D}$ of labeled documents (see, e.g. [21]).

2.2 Hidden Markov models

HMMs have been introduced several years ago as a tool for probabilistic sequence modeling. The interest in this area developed particularly in the Seventies, within the speech

[1] A related formulation would consist of assigning a global category to a whole multi-page document, but this formulation is not considered in this paper.

[2] A Bayesian network is an annotated graph in which nodes represent random variables and *missing* edges encode conditional independence statements amongst these variables. Given a particular state of knowledge, the semantics of a Bayesian networks determine whether collecting evidence about a set of variables does modify one's belief about some other set of variables [24, 11].

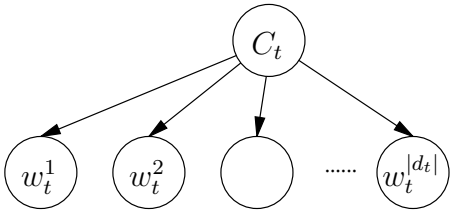

Figure 1: Bayesian network for the Naive Bayes classifier.

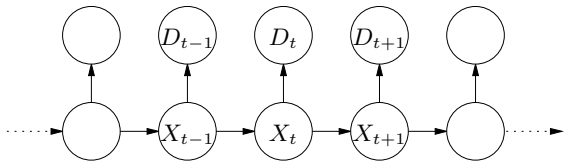

Figure 2: Bayesian networks for standard HMMs.

recognition research community [25]. During the last years a large number of variants and improvements over the standard HMM have been proposed and applied. Undoubtedly, Markovian modeling is now regarded as one of the most significant state-of-the-art approaches for sequence learning. Besides several applications in pattern recognition and molecular biology, HMMs have been also applied to several text related tasks, including natural language modeling [5] and, more recently, information retrieval and extraction [9, 20]. The recent view of the HMM as a particular case of Bayesian networks [2, 19, 27] has helped the theoretical understanding and the ability to conceive extensions to the standard model in a sound and formally elegant framework.

An HMM describes two related discrete-time stochastic processes. The first process pertains to hidden discrete state variables, denoted X_t, forming a first-order Markov chain and taking realizations on a finite alphabet $\{x^1, \cdots, x^N\}$. The second process pertains to observed variables or *emissions*, denoted D_t. Starting from a given state at time 0 (or given an initial state distribution) the model probabilistically transitions to a new state X_1 and correspondingly emits observation D_1. The process is repeated recursively until an end state is reached. Note that, as this form of computation may suggest, HMMs are closely related to stochastic regular grammars [5]. The Markov property prescribes that X_{t+1} is conditionally independent of $X_1, \ldots, X_{t-1}$ given X_t. Furthermore, it is assumed that D_t is independent of the rest given X_t. These two conditional independence assumptions are graphically depicted using the Bayesian network of Figure 2. As a result, an HMM is fully specified by the following conditional probability distributions[3]:

$$\begin{array}{ll} P(X_t|X_{t-1}) & \text{(transition distribution)} \\ P(D_t|X_t) & \text{(emission distribution)} \end{array} \quad (3)$$

Since the process is stationary, the transition distribution can be represented as a square probability matrix whose entries are transition probabilities $P(X_t = x^i|X_{t-1} = x^j)$, abbreviated as $P(x^i|x^j)$ in the following. In the classic literature, emissions are restricted to symbols in a finite alphabet or multivariate continuous variables [25]. As explained in the next section, our model allows emissions to be bag-of-words.

3. THE MULTI-PAGE CLASSIFIER

We now turn to the description of our classifier for multi-page documents. This section presents the architecture and the algorithms for grammar extraction, training, and classification.

3.1 Architecture

In our case, HMM emissions are associated with entire pages of the document. Thus the realizations of the observation D_t are bag-of-words representing the text in the t-th page of the document. Within our framework, states are related to pages categories by a a deterministic function ϕ that maps state realizations into page categories. We assume that ϕ is a surjection but not a bijection, i.e. that there are more state realizations than categories. This enriches the expressive power of the model, allowing different transition behaviors for pages of the same class, depending on where the page is actually encountered within the sequence. However, if the page *contents* depends on the category but not on the context of the category within the sequence[4], the use of multiple states per category may introduce too many free parameters and it may be convenient to assume that

$$P(D_t|x^i) = P(D_t|x^j) = P(D_t|c^k) \;\text{ if }\; \phi(x^i) = \phi(x^j) = c^k. \quad (4)$$

This assumption constrains emission parameters to be the same for all the HMM states labeled by the same page category, a form of parameters sharing that may help to reduce overfitting. The emission distribution is then defined as for the Naive Bayes classifier, i.e. for every observed page d_t

$$P(d_t|c^k) = \prod_{i=1}^{|d_t|} P(w_t^i|c^k) \quad (5)$$

Therefore, the architecture can be graphically described as the merging of the Bayesian networks for HMMs and Naive Bayes, as shown in Figure 3. We remark that the state (and hence the category) at page t depends not only on the contents of the page, but also on the contents of other pages in the document. This probabilistic dependency implements

[3]We adopt the standard convention of denoting variables by uppercase letters and realizations by the corresponding lowercase letters. Moreover, we use the table notation for probabilities as in [11]; for example $P(X)$ is a shorthand for the table $[P(X{=}x^1), \ldots, P(X{=}x^r)]$ and $P(X, y|Z)$ denotes the two-dimensional table with entries $P(X{=}x^i, Y{=}y|Z{=}z^k)$.

[4]Of course this does not mean that the *category* is independent on the context.

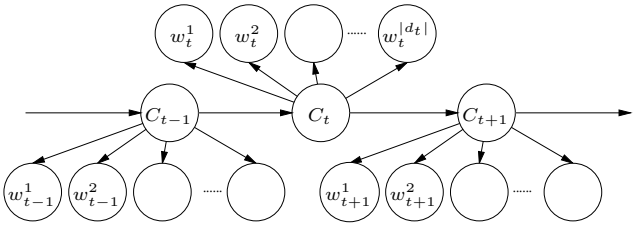

Figure 3: Bayesian network for the hybrid HMM Naive Bayes architecture.

the mechanism for taking contextual information into account.

The algorithms used in this paper are derived from the literature on Markov models [25], inference and learning in Bayesian networks [24, 11, 10], and classification with Naive Bayes [17, 15]. In the following we sketch the main issues related to the integration of all these methods.

3.2 Induction of HMM topology

The *structure* or topology of an HMM is a representation of the allowable transitions between hidden states. More precisely, the topology is described by a directed graph whose vertices are state realizations $\{x^1, \ldots, x^N\}$, and whose edges are the pairs (x^j, x^i) such that $P(x^i|x^j) \neq 0$. An HMM is said to be *ergodic* if its transition graph is fully-connected. However, in almost all interesting application domains, less connected structures are better suited for capturing the observed properties of the sequences being modeled, since they convey domain prior knowledge. Thus, starting from the right structure is an important problem in practical hidden Markov modeling. As an example, consider Figure 4, showing a (very simplified) graph that describes transitions between the parts of a hypothetical set of books. Possible state realizations are[5] {start, title, dedication, preface, toc, regular, index, end }. The structure indicates, among other things, that only dedication, preface, or table of contents can follow the title page. Self-loops indicate that a given category can be repeated for several consecutive pages. While

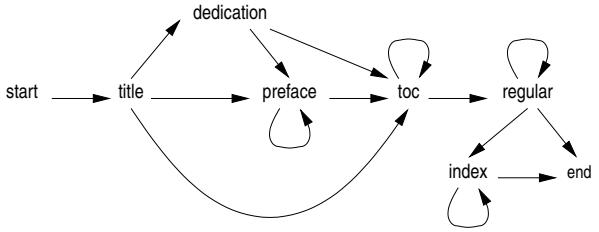

Figure 4: Example of HMM transition graph.

a structure of this kind could be hand-crafted by a domain expert, it is may be more advantageous to learn it automatically from data.

We now briefly describe the solution adopted to automatically infer HMM transition graphs from sample multi-page documents. Let us assume that all the pages of the available training documents are labeled with the class they belong to. One can then imagine to take advantage of the observable distribution of data to search for an effective structure in the space of HMMs topologies. Our approach is based on the application of an algorithm for data-driven model induction adapted from previous works in Bayesian HMM induction [28] and construction of HMMs of text phrases for information extraction [20]. The algorithms starts by building a structure that is capable only to "explain" the available training sequences (a maximally specific model). The initial structure includes as many paths (from the initial state to the final one) as there are training sequences. Every path is associated with one sequence of pages, i.e. a distinct state is created for every page in the training set. Each state x is labeled by $\phi(x)$, the category of the corresponding page in the document. Note that, unlike the example shown in Figure 4, several states are generated for the same category. The algorithm then iteratively applies merging heuristics that collapse states so as to augment generalization capabilities over unseen sequences. The first heuristic, called neighbor-merging, collapse two states x and x' if they are neighbors in the graph and $\phi(x) = \phi(x')$. The second heuristic, called V-merging, collapses two states x and x' if $\phi(x) = \phi(x')$ and they share a transition from or to a common state, thus reducing the branching factor of the structure.

3.3 Inference and learning

Given the HMM topology extracted by the algorithm described above, the learning problem consists of determining transition and emission parameters. One important distinction that need to be made when training Bayesian network is whether or not all the variables are observed. Assuming complete data (all variables observed), maximum likelihood estimation of the parameters could be solved using a one-step algorithm that collects sufficient statistics for each parameter [10]. In our case, data are complete if and only if the following two conditions are met:

1. there is a one-to-one mapping between HMM states and page categories (i.e. $N = K$ and for $k = 1, \ldots, N$, $\phi(x^k) = c^k$), and
2. the category is known for each page in the training documents, i.e. the dataset consists of sequences of pairs $(\{d_1, c_1^*\}, \ldots, \{d_T, c_T^*\})$, c_t^* being the (known) category of page t and T being the number of pages in the document.

Under these assumptions, estimation of transition parameters is straightforward and can be accomplished as follows:

$$P(x^i|x^j) = \frac{N(c^i, c^j)}{\sum_{\ell=1}^{N} N(c^\ell, c^j)} \tag{6}$$

where $N(c^i, c^j)$ is the number of times a page of class c^i follows a page of class c^j in the training set. Similarly, estimation of emission parameters in this case would be accomplished exactly like in the case of the Naive Bayes classifier (see, e.g. [21]):

$$P(w^\ell|c^k) = \frac{1 + N(w^\ell, c^k)}{|V| + \sum_{j=1}^{|V|} N(w^j, c^k)} \tag{7}$$

[5]Note that in this simplified example ϕ is a one-to-one mapping.

where $N(w^\ell, c^k)$ is the number of occurrences of word w^ℓ in pages of class c^k and $|V|$ is the vocabulary size ($1/|V|$ corresponds to a Dirichlet prior over the parameters and plays a regularization role for whose words which are very rare within a class).

Conditions 1 and 2 above, however, are normally not satisfied. First, in order to model more accurately different contexts in which a category may occur, it may be convenient to have multiple distinct HMM states for the same page category. Second, labeling pages in the training set is a time consuming process that needs to be performed by hand and it may be important to use also unlabeled documents for training [13, 23]. This means that label c_t^* may be not available for some t. If assumption 2 is satisfied but assumption 1 is not, we can derive the following approximated estimation formula for transition parameters:

$$P(x^i|x^j) = \frac{N(x^i, x^j)}{\sum_{\ell=1}^{N} N(x^\ell, x^j)} \tag{8}$$

where $N(x^i, x^j)$ counts how many times state x^i follows x^j during the state merge procedure described in Section 3.2. However, in general, the presence of hidden variables requires an *iterative* maximum likelihood estimation algorithm, such as gradient ascent or expectation-maximization (EM). Our implementation uses the EM algorithm, originally formulated in [7] and usable for any Bayesian network with local conditional probability distributions belonging to the exponential family [10]. Here the EM algorithm essentially reduces to the Baum-Welch form [25] with the only modification that some evidence is entered into state variables. State evidence is taken into account in the E-step by changing forward propagation as follows:

$$\alpha_t(j) = \begin{cases} 0 & \text{if } \phi(x^j) \neq c_t^* \\ \sum_{i=1}^{N} \alpha_{t-1}(i) P(x^j|x^i) P(d_t|x^j) & \text{otherwise} \end{cases} \tag{9}$$

where $\alpha_t(i) = P(d_1 d_2 \cdots d_t, X_t = x^i)$ is the forward variable in the Baum-Welch algorithm.

The M-step is performed in the standard way for transition parameters, by replacing counts in Equation 6 with their expectations given all the observed variables. Emission probabilities are also estimated using expected word counts. If parameters are shared as indicated in Equation 4, these counts should be summed over states having the same label. Thus in the case of incomplete data, Equation 7 is replaced by

$$P(w^\ell|c^k) = \frac{S + \sum_p \sum_t N(w^\ell, c^k) \sum_{i:\phi(x^i)=c^k} P(x^i|d_t)}{S|V| + \sum_{j=1}^{|V|} \sum_p \sum_t N(w^j, c^k) \sum_{i:\phi(x^i)=c^k} P(x^i|d_t)}$$

where S is the number of training sequences, $N(w^\ell, c^k)$ is the number of occurrences of word w^ℓ in pages of class c^k, and $P(x^i|d_t)$ is computed by the Baum-Welch procedure during the E-step. The sum on p extends over training sequences, while the sum on t extends over pages of the p-th document in the training set. The E- and M-steps are iterated until a local maximum of the (incomplete) data likelihood is reached.

It is interesting to point out a related application of the EM algorithm for learning from labeled and unlabeled documents [23]. In that paper the only concern was to allow the learner to take advantage of unlabeled documents in the training set. As a major difference, the method in [23] assumes flat single-page documents and, if applied to multi-page documents, would be equivalent to a zero-order Markov model that cannot take into account contextual information.

3.4 Page classification

Given a document of T pages, classification is performed by first computing the sequence of states $\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_T$ that was most likely to have generated the observed sequence of pages, and then mapping each state to the corresponding category $\phi(\hat{x}_t)$. The most likely state sequence can be obtained by running the an adapted version of Viterbi's algorithm, whose more general form is the max-propagation algorithm for Bayesian networks described in [11].

3.5 Feature selection

Text pages should be first preprocessed with common information retrieval techniques, including stemming and stop words removal. Still, the bag-of-words representation of pages can lead to a very high-dimensional feature space corresponding to the vocabulary extracted from training documents. A high-dimensional feature space, especially in this case where features are noisy because of OCR errors, may lead to the overfitting phenomenon: the learner has very high accuracy on the training set but generalization to new examples is poor. Feature selection is a technique for limiting overfitting by removing non-informative words from documents. In our experiments we performed feature selection using information gain [30]. This criterion is often employed in different machine learning contexts. It measures the average number of bits of information about the category that are gained by including a word in a document. For each dictionary term w, the gain is defined as

$$\begin{aligned} G(w) &= -\sum_{k=1}^{K} P(c^k) \log_2 P(c^k) \\ &+ P(w) \sum_{k=1}^{K} P(c^k|w) \log_2 P(c^k|w) \\ &+ P(\overline{w}) \sum_{k=1}^{K} P(c^k|\overline{w}) \log_2 P(c^k|\overline{w}) \end{aligned}$$

where $\overline{w}$ denotes the absence of word w. Feature selection is performed by retaining only the words having the highest average mutual information with the class variable. OCR errors, however, can produce very noisy features which may be responsible of poor performance even if feature selection is performed. For this reason, it may be convenient to prune from the dictionary (before applying the information gain criterion) all the words occurring in the training set with a frequency below a given threshold h.

3.6 Learning with labeled and unlabeled pages

Creating a training set for text categorization involves hand labeling in order to assign a category to each document. Since this is an expensive human activity, it is interesting to evaluate a classification system when only a fraction of the training documents pages are labeled, while other

documents are used without a category label. Clearly, unlabeled documents are available at very low cost. In the case of isolated page classification, previous research has demonstrated that learners such as Naive Bayes and support vector machines can take advantage of the inclusion in the training set of documents whose class is unknown [13, 23]. In particular, the method presented in [23] uses EM to deal with unobserved labels.

In the case of multi-page documents, the presence of missing labels means that some pages of the training document sequences have no assigned category. The architecture introduced in this paper (see Figure 3) can easily handle the presence of unlabeled pages in the training set. Basically, evidence is entered into the states of the HMM chain only for those pages for which a label is known, while other state variables are left unobserved. The belief propagation algorithm is in charge of computing probabilities for these hidden variables.

However, the structure learning algorithm presented in Section 3.2 cannot be applied in the case of partially labeled documents. Instead, it is possible to use ergodic (fully connected) HMMs and deriving a transition structure by pruning, after the learning phase, those transitions having small probabilities with respect to an assigned threshold. In this way, we let EM derive a specific structure for the model (note that the only alternative in the case of partially labeled documents would be to obtain a transition graph from a domain expert).

4. EXPERIMENTAL RESULTS

A preliminary evaluation of our system has been conducted in a digital library domain where data are naturally organized in the form of page sequences. The main purpose of our experiments was to make a comparison between our multi-page classification approach and a traditional isolated page classification system.

4.1 Data Set

We have chosen to evaluate the model over a subset of the Making of America (MOA) collection, a joined project between the University of Michigan and Cornell University (see `moa.umdl.umich.edu/about.html` and [26]) for collecting and making available digitized books and periodicals about history and evolution processes of the American society between the XIX and XX century. Presently, the whole archive contains electronic versions of important magazines of the XIX century. In our experiments, we selected a subset of the journal *American Missionary* (AMis), a sociological magazine with strong Christian guidelines. The task consists of correctly classifying pages of previously unseen documents into one of the ten categories described in Table 1. Most of these categories are related to the topic of the articles, but some are related to the parts of the journal (i.e. Contents, Receipts, and Advertisements). The dataset we selected contains 95 issues from 1884 to 1893, for a total of 3222 OCR text pages. Special issues and final report issues (typically November and December issues) have been removed from the dataset as they contain categories not found in the rest. The first year was selected as training set (10 training sequences, 342 pages). The remaining documents (from 1885 to 1993, for a total of 2880 pages) were used as a test set. The ten categories are temporally stable over the 1883–1893 time period.

Name	Description
1. Contents	Cover and index of surveys
2. Editorial	Editorial articles
3. The South	Afro-Americans' survey
4. The Indians	American Indians' survey
5. The Chinese	Reports from China missions
6. Bureau of Women's Work	Female conditions
7. Children's Page	Education and childhood
8. Communications	Magazine informations
9. Receipts	Lists of founders
10. Advertisements	contents is mostly graphic

Table 1: Categories in the *American Missionary* domain.

Category labels were obtained semi-automatically, starting from the MOA XML files supplied with the documents collection. The assigned category was then manually checked. In the case of pages containing the end and the beginning of two articles belonging to different categories, the page was assigned the category of the ending article.

Each page within a document is represented as a Bag-of-Words, counting the number of word occurrences within the page. It is worth remarking that in this application instances are text documents obtained by an OCR process. Imperfections of recognition algorithm and the presence of images in some pages yields noisy text, containing misspelled or nonexistent words, and trash characters (see [3] for a report of OCR accuracy in the MOA digital library). Although these errors may negatively affect the learning process and subsequent results in the evaluation phase, we made no attempts to correct and filter out misspelled words, except for the feature selection process described above. However, since OCR extracted documents preserve the text layout found in the original image, it was necessary to rejoin words that had been hyphenated due to line breaking.

4.2 Feature selection and isolated page classification

The purpose of the experiments in this section is to investigate the effects of feature selection and to assess the baseline prediction accuracy that can be attained using the Naive Bayes classifier on isolated pages. In a set of preliminary evaluations we have found that best performance are achieved by pruning words with less than $h = 10$ occurrences and then selecting an optimal set of informative words. We performed several tests by changing the information gain threshold that determines if a word is sufficiently informative (see Section 3.5), resulting in different vocabulary sizes with different accuracy of prediction. For each reduced vocabulary size we ran the Naive Bayes classifier on isolated pages. Results are shown in Figure 5. Vocabulary size ranges from 15635 words (no feature selection), yielding 65.07% classification accuracy, to 25 words, yielding 53.16% accuracy. The optimal vocabulary size is 297 words, obtained with a threshold gain of 0.089, yielding the best test-set accuracy of 72.57%. This result (72.57%) was considered as the base measure for performance comparison between our model and the Naive Bayes classifier.

4.3 Sequential page classification

Using the hybrid model presented in Section 3, documents can be organized into ordered sequences of pages. The training set contains 10 sequences (monthly issues) of the same

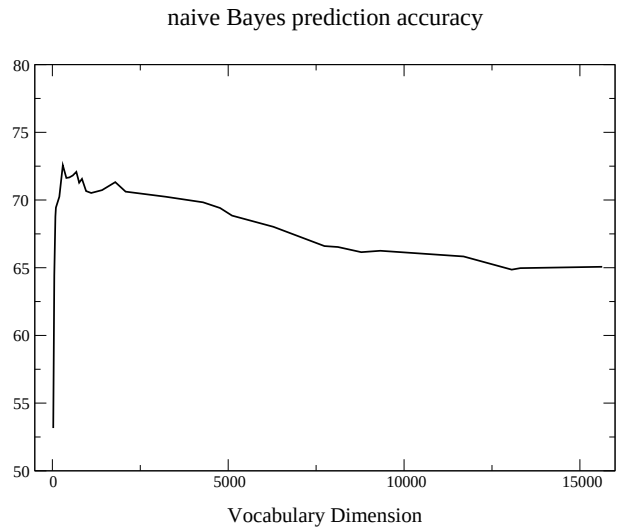

Figure 5: Naive Bayes accuracy as a function of vocabulary size (information gain criterion). Optimal vocabulary size is 297 words.

Category	Sequential	Isolated	Error red.
Contents	**100**	100	0%
Editorial	**80.9**	63.11	48.2%
The South	**90.81**	71.84	67.4%
The Indians	**61.07**	44.3	30.1%
The Chinese	**69.93**	60.78	23.3%
Bureau W.W.	**74.73**	66.3	25.0%
Children's Page	**78.26**	45.65	60.0%
Communications	**93.55**	92.47	14.3%
Receipts	**98.31**	98.31	0%
Advertisements	**90.7**	62.79	75.0%
Total Accuracy	**85.28**	72.57	46.3%

Table 2: Isolated classification (using the best Naive Bayes) vs. sequential classification (using the hybrid HMM with model merging).

342 documents for year 1884, while test set is organized into 85 sequences for a total of 2880 documents from year 1885 to 1893. The bag-of-words representation of pages fed into the HMM classifier was identical to that previously used with Naive Bayes (including preprocessing and feature selection with a vocabulary of 297 words). We have considered two settings for validating the system. In the first setting, it is assumed that category labels c_t^* are available for all the pages in the training set. In the second setting, some category labels are held out and training uses labeled and unlabeled pages.

4.3.1 Completely labeled documents

In the case of completely labeled documents it is possible to run the structure learning algorithm presented in Section 3.2. Figure 6 reports the structure learned from the 10 training issues. Each vertex in the transition graph is associated with one HMM state and is labeled with the corresponding category (see Table 1). Edges are labeled with the transition probability from source to target state, computed by counting state transitions during the state merging procedure (see Equation 8). The associated stochastic

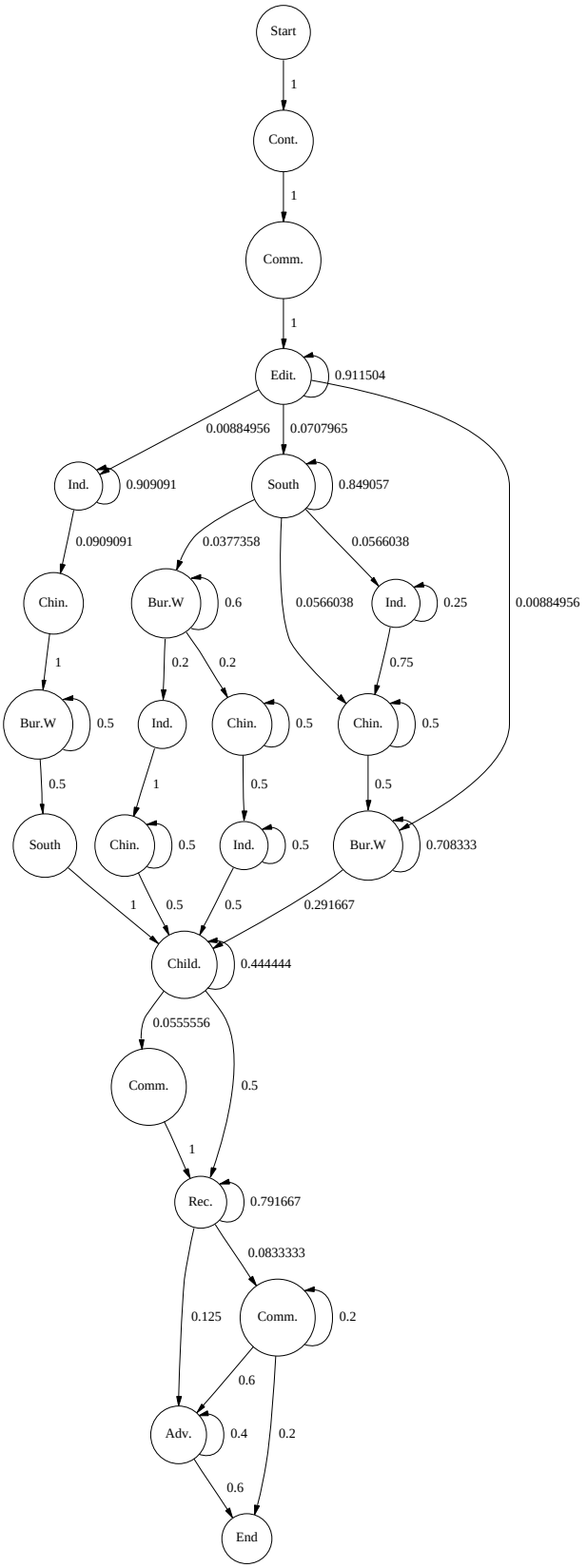

Figure 6: Data induced HMM topology for American Missionary, year 1884.

grammar implies that valid AMis sequences ought to start with the index page (class "Contents"), followed by a page of general communications. Next state is associated with a page of an editorial article. Self transition here has a value of 0.91, meaning that with high probability the next page will belong to the editorial too. With lower probability (0.07) next page is one of the "The South" survey or (prob. 0.008) "The Indians" or "Bureau of Women's work". Continuing this way we can associate a probability to each string of page categories. Since our purpose is to predict the correct string of categories, a good grammar helps filtering out classification hypothesis which generate low (or zero) probability strings. Note that under the parameter sharing assumption (see Equation 4), once the HMM structure is given, an estimate of the emission probabilities can be obtained using Equation 7. These values can be plugged in as initial emission parameters for the EM algorithm. Classification is finally performed by computing the most likely state sequence.

Table 2 summarizes classification results on test set documents sequences, after a training phase applied both to Naive Bayes and our hybrid model. We report accuracy of prediction on single classes and average accuracy over the total of text documents. Comparison is made with respect to the best isolated-page classifier. The hybrid HMM classifier (performing sequential classification) achieves 85.28% accuracy and consistently outperforms the plain Naive Bayes classifier working on isolated pages. The relative error reduction is about 46%, i.e. roughly half of the errors are recovered thanks to contextual information. In particular, it is interesting to note the large error reduction for the category "Advertisements." Pages in this category typically contain several images and few words of text. The isolated page classifier is subject to prediction errors in this case since parameter estimation for rarely occurring words can be poor. On the other hand, the constraints imposed by the grammar allow to recover many prediction errors since advertisements normally occur near the end of each issue.

In Figure 7 we report classification performances of the hybrid model on single issues of the journal. The graph is to be interpreted as the classifier temporal trend from 1885 to 1894. Negative accuracy peaks correspond to test issues with more than 70 pages, a significant deviation from the average number of pages per issue (about 32). Values range from a minimum of 50% to a maximum of 97.09% with 10.41 as standard deviation. To visualize a smoother trend, we calculated a running average over a temporal window of 10 months, showing a clear superior trend over standard naive Bayes.

4.3.2 *Partially labeled documents*

We have performed six different experiments, for different percentages of labeled documents. In this case the structure learning algorithm cannot be applied and we used ergodic HMMs with ten states (one per class). After training, transition with probabilities $< 10^{-3}$ were pruned. In one of the six experiments we used all the available page labels with an ergodic HMM. This experiment is useful to provide a basis for evaluating the benefits of the structure learning algorithm presented in Section 3.2.

Table 3 shows detailed results of the experiments. Classification accuracy is shown for single classes and for the the entire test set. As we can see, EM being completed uninformed

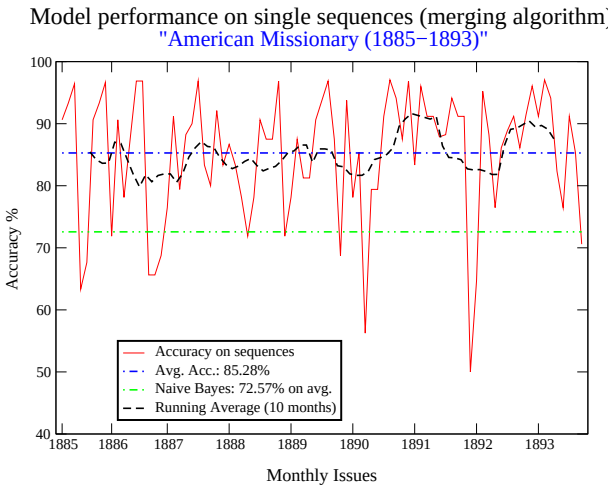

Figure 7: Performance of the hybrid model on single sequences (merging algorithm).

	% of labeled documents					
Category	0	30	50	70	90	100
Contents	0	100	100	100	100	100
Editorial	20.76	21.12	59.67	58.6	67.62	71.41
South	1.51	83.58	69.73	84.94	84.34	84.19
Indians	10.07	0	55.03	51.68	50.34	58.39
Chinese	0	27.45	83.66	76.47	75.82	75.16
Bur.W.W.	0	43.22	63.74	63	64.84	65.93
Child. P.	4.35	78.26	73.91	58.7	78.27	76.09
Commun.	0	91.4	91.4	93.55	93.55	93.55
Receipts	0	89.27	98.68	97.36	98.31	98.31
Advert.	81.4	69.77	93.02	90.7	90.7	90.7
Total Accuracy	8.23	55.66	73.54	75.66	78.7	80.24

Table 3: Results achieved by the model trained by Expectation-Maximization, varying percentage of labeled documents.

(0% evidence) is worse than the random guess (8.23% accuracy). With 50% of labeled documents, the model outperforms Naive Bayes (73.54% against 72.57%). This is a positive result, because the Naive Bayes training phase (in the standard formulation) need the knowledge of all document labels, while in this setting we simulate the knowledge of only a half of them. With greater percentages of labeled documents, performances begin to saturate reaching a maximum of 80.24% when all the labels are known. This result is worse compared to the 85.28% obtained with the first strategy (see Section *4.3.1*). The main difference is that in this case we started training from an ergodic model and we used one state per class. This confirms that in the case of completely labeled documents it is advantageous to use more states per class and to use the data-driven algorithm for structure selection.

5. CONCLUSIONS

We have presented a text categorization system for multi-page documents which is capable of effectively taking into account contextual information to improve accuracy with respect to traditional isolated page classifiers. Our method can smoothly deal with unlabeled pages within a document,

although we have found that learning the HMM structure further improves performance compared to starting from an ergodic structure. The system uses OCR extracted words as features. Clearly, richer page descriptions could be integrated in order to further improve performance. For example, optical recognizer output information about the font, size, and position of text, that may be important to help discriminating between classes. Moreover, OCR text is noisy and another direction for improvement is to include more sophisticated feature selection methods, like morphological analysis or the use of n-grams [4, 14].

Another aspect is the granularity of document structure being exploited. Working at the level of pages is straightforward since page boundaries are readily available. However, actual category boundaries may not coincide with page boundaries and some pages contains portions of text related to different categories. Although this is not very critical for single-column journals such as the *American Missionary*, the case of documents typeset in two or three columns certainly deserves attention. A further direction of investigation is therefore related to the development of algorithms capable of performing automatic segmentation of a continuous stream of text, without necessarily relying on page boundaries.

The categorization method presented in this paper is targeted to textual information. However, the same hybrid HMM methodology could be applied for classification of pages based on layout information, provided an adequate emission model is available. A suitable generative model for document layout is presented in [8].

Finally, categorization algorithms that includes contextual information may be very useful for other types of documents natively available in electronic form. For example, the categorization of web pages may take advantage of the contents in neighbor pages (as defined by the hyperlink structure of the web).

6. ACKNOWLEDGMENTS

We thank Oya Rieger and the Cornell University Library for providing the data collected within the Making of America project. This research was partially supported by EC grant # IST-1999-20021 under METAe project.

7. REFERENCES

[1] The metadata engine project. `http://meta-e.uibk.ac.at`, 2001.

[2] Y. Bengio and P. Frasconi. An input output HMM architecture. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 427–434. MIT Press, 1995.

[3] D. A. Bicknese. Measuring the accuracy of the OCR in the Making of America. Report available at `moa.umdl.umich.edu/moaocr.html`, 1998.

[4] W. Cavnar and J. Trenkle. N-Gram based text categorization. In *Prof. of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, NV, 1994.

[5] E. Charniak. *Statistical Language Learning*. MIT Press, 1993.

[6] W. W. Cohen. Text categorization and relational learning. In *Proceedings of the Twelfth International Conference on Machine Learning*, Lake Tahoe, California, 1995.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39:1–38, 1977.

[8] M. Diligenti, P. Frasconi, and M. Gori. Image document categorization using hidden tree-Markov models and structured representations. In S. Singh, N. Murshed, and W. Kropatsch, editors, *Second Int. Conf. on Advances in Pattern Recognition*, volume 2013 of *LNCS*. Springer, 2001.

[9] D. Freitag and A. McCallum. Information extraction with hmm structures learned by stochastic optimization. In *Proc. 12th AAAI Conference*, Austin, TX, 2000.

[10] D. Heckerman. Bayesian networks dor data mining. *Data Mining and Knowledge Discovery*, 1(1):79–120, 1997.

[11] F. Jensen. *An Introduction to Bayesian Networks*. Springer Verlag, New York, 1996.

[12] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*. Springer, 1998.

[13] T. Joachims. Transductive inference for text classification using support vector machines. In *Int. Conf. on Machine Learning*, 1999.

[14] M. Junker and R. Hoch. Evaluating OCR and non-OCR text representations for learning document classifiers. In *Prof. ICDAR 97*, 1997.

[15] T. Kalt. A new probabilistic model of text classification and retrieval. CIIR TR98-18, University of Massachusetts, 1996. url: `ciir.cs.umass.edu/publications/`.

[16] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proc. Fourteenth Int. Conf. on Machine Learning*, 1997.

[17] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *SIGIR-94*, 1994.

[18] D. Lewis and M. Ringuette. Comparison of two learning algorithms for text categorization. In *Proc. 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994.

[19] H. Lucke. Bayesian belief networks as a tool for stochastic parsing. *Speech Communication*, 16:89–118, 1995.

[20] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000.

[21] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[22] H. Ng, W. Goh, and K. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proc. of the 20th Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 67–73, 1997.

[23] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[24] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[25] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[26] E. Shaw and S. Blumson. Online searching and page presentation at the University of Michigan. *D-Lib Magazine*, July/August 1997. url: `www.dlib.org/dlib/july97/america/07shaw.html`.

[27] P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9(2):227–269, 1997.

[28] A. Stolcke and S. Omohundro. Hidden Markov Model induction by bayesian model merging. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 11–18. Morgan Kaufmann, San Mateo, CA, 1993.

[29] Y. Yang and C. Chute. An example-based mapping method for text classification and retrieval. *ACM Transactions on Information Systems*, 12(3):252–277, 1994.

[30] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.