A Hybrid Machine Learning Approach for Information Extraction

Eduardo F. A. Silva, Flavia A. Barros, Ricardo B. C. Prudêncio
Center of Informatics
Federal University of Pernambuco
Pobox 7851 - CEP 50732-970 - Recife (PE) - Brazil
efas, fab@cin.ufpe.br, prudencio.ricardo@gmail.com

Abstract

Information Extraction (IE) aims to extract from textual documents only the relevant data required by the user. In this paper, we propose a hybrid machine learning approach for IE on semi-structured texts that combines conventional text classification techniques and Hidden Markov Models (HMM). In this approach, a text classifier technique generates an initial output, which is refined by an HMM, providing a globally optimal extraction. An implemented prototype was used to extract information from bibliographic references, reaching a consistent gain in performance through the use of the HMM.

1 Introduction

Considering the huge amount of textual documents available in digital repositories, it is of great interest to build systems capable of automatically retrieving only the data which interests a user. *Information Extraction* (IE) systems arise as a means to facilitate the information access, by extracting from the documents only the parts that correctly fill in a set of pre-defined output slots (fields).

Among the approaches for IE, we highlight the use of Machine Learning (ML) algorithms as text classifiers [5]. Here, the document is initially divided into fragments which will be later associated to the output slots by an ML algorithm. The classification is performed based on descriptive features of the fragment (e.g. its length, presence of terms, etc). Despite their advantages, these systems classify each input fragment independently on the other fragments. As such, they miss important information about the document's structure [2].

In order to minimize the above dificulty, we propose a hybrid IE approach for semi-structured texts which combines traditional ML text classifiers with Hidden Markov Models (HMM) [7]. In this approach, a ML text classifier generates an initial classification of the input text fragments, which is refined by the HMM. The HMMs are able to take into account dependencies among the input fragments, thus favoring a globally optimal classification for the whole input sequence [2].

As case study, we implemented a prototype for the domain of bibliographic references, aiming to extract information such as author, title, year, etc. A reference is seen as a semi-structured text with a high variance in its structure [2]. The experiments performed with the prototype revealed a consistent gain in the performance with the use of the HMM, ranging from 3.80 to 22.54 percentile points.

Section 2 presents techniques to IE. Section 3 details the proposed solution. Section 4 describes the implemented prototype. Section 5 brings the experiments and obtained results. Finally, section 6 presents some conclusions.

2 Information Extraction

Information Extraction (IE) is concerned with extracting relevant data from a collection of documents [1]. An IE system identifies document fragments that correctly fill in slots in a given output form. The extracted data can be directly presented to the user or stored in a database.

Machine Learning (ML) techniques have been largely used for IE in order to automatically generate extraction rules from tagged corpora [1]. Among the ML systems for IE, we cite those based on the learning of finite automata [4] and regular expressions [8]. Systems based on these techniques represent rules using symbolic languages that are easier to interpret. However, they require regular patterns or clear text delimiters [8] and hence are less adequate for texts which show a higher degree of variation in structure.

An alternative approach for IE is the use of conventional ML algorithms[1] as text classifiers [5, 3]. Initially, the input text is divided into fragments which will be later associated to the output slots. Next, an ML algorithm classifies each

[1] Conventional ML algorithms may be for instance the Naive Bayes classifier and the kNN algorithm.

Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (HIS'06)
0-7695-2662-4/06 $20.00 © 2006 IEEE

fragment based on its descriptive features (e.g., number of words, occurrence of numbers, etc). Here, the class values correspond to the slots in the output form. The major drawback with these systems is that they perform a local and independent classification for each fragment, thus overlooking relevant structural information present in the document.

With the aim of minimizing the above difficulty, a number of researchers have used HMMs for IE [5][2]. HMMs are able to take into account dependencies among the input fragments, thus maximizing the probability of a globally optimal classification for the whole input sequence. Here, each slot (class) to be extracted is associated to a hidden state. Given a sequence of input fragments, the HMM determines the most probable sequence of slots associated to the input sequence. Despite their advantages, the HMMs can only consider one feature of each fragment (e.g., size or position) [3], which may compromise local classification.

3 The Proposed Approach

We propose here a hybrid approach for IE on semi-structured texts in which an initial extraction performed by a conventional text classifier is refined through the use of an HMM. By combining these techniques, we safeguard their advantages while overcoming their limitations. As mentioned, conventional text classifiers offer a locally optimal classification for each input fragment, however disregarding the relationships among fragments. On the other hand, HMMs offer a globally optimal classification for all input fragments, but are not able to treat multiple features of fragments.

Figure 1 presents the proposed approach, illustrated in the domain of bibliographic references. As it can be seen, the IE process consists of the following main steps:

1. *Phase 1 - Extraction using a conventional text classifier.* The initial extraction process is divided into:

   (a) *Fragmentation of the input text.* The input text must be divided into candidate fragments for filling in the output slots. This segmentation is commonly performed by a set of heuristics that may consider text delimiters.

   (b) *Feature extraction.* A vector of features is created for describing each fragment and is used in the classification of the fragment.

   (c) *Fragment classification.* A classifier decides which output slot will be filled in by each input fragment. Here, we build conventional ML algorithms by using a corpus of tagged fragments as training set.

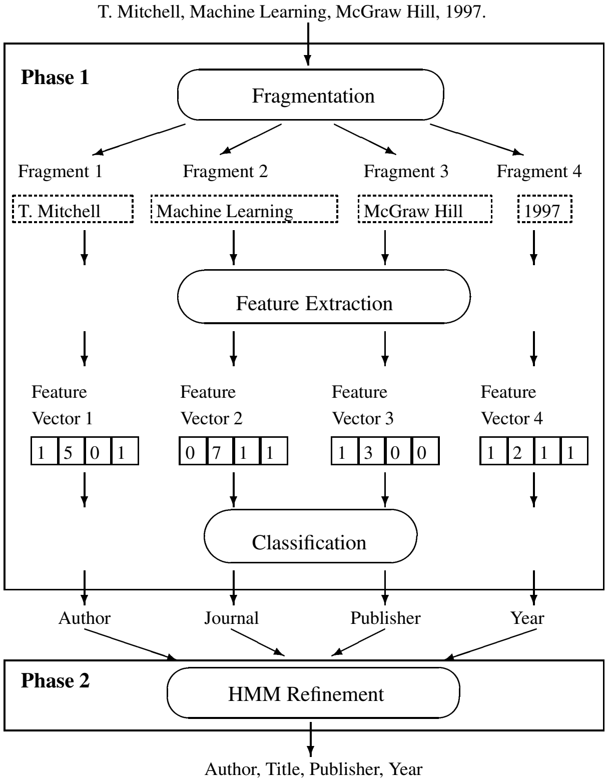

Figure 1. Proposed Approach

2. *Phase 2 - Refinement of the results using an HMM.* The HMM refines the initial extraction, providing a globally optimal classification for the whole sequence of input fragments.

An HMM is a probabilistic finite automata that consists of: (1) a set of hidden states S; (2) a transition probability distribution in which $Pr[s'/s]$ is the probability of making a transition from the hidden state $s \in S$ to $s' \in S$; (3) a finite set of symbols T emitted by the hidden states; and (4) an emission probability distribution in which $Pr[t/s]$ is the probability of emitting the symbol $t \in T$ in state $s \in S$. The Viterbi algorithm is used in the classification process, delivering a sequence of hidden states with the highest probability of generating each input sequence of symbols. The HMM may induces the probability distributions $Pr[s'/s]$ and $Pr[t/s]$ by the use of a training set that associates hidden states and emitted symbols.

Here, each hidden state represents an output slot, and the emitted symbols represent the classes predicted by Phase 1. Formally, let $\mathcal{C} = \{c_1, \ldots, c_K\}$, where each $c_k \in \mathcal{C}$ represents a different slot in the output form. The set of hidden states is defined here as $S = \{s_1, \ldots, s_K\}$ in such way that there is a one-to-one mapping between hidden states and class values. If the correct class of the j-th fragment is $c_k \in \mathcal{C}$, then the j-th state of the HMM is s_k. Similarly, the set of symbols is defined as $T = \{t_1, \ldots, t_K\}$, in such a way that, if the prediction of the Phase 1 for the j-th fragment is c_k then the j-th emitted symbol is t_k.

The transition probability $Pr[s_{k_1}|s_{k_2}]$ between the states s_{k_1} and s_{k_2} actually represents the probability that the correct class of a fragment is c_{k_1} given that the correct class of the previous fragment in the input text is c_{k_2}. The emission probability $Pr[t_{k_1}|s_{k_2}]$, in turn, represents the probability that the classifier of Phase 1 predicts the class value c_{k_1}, given that the correct class of the fragment is c_{k_2}.

Each training example consists of a list of pairs containing a *symbol* (i.e., the class predicted to a specific fragment in Phase 1) and the associated *hidden state* (i.e., the class to which the fragment actually belongs). The HMM takes as input the whole sequence of class values provided by Phase 1 and returns a refined classification for the given fragments.

4 Case Study: IE on Bibliographic References

As case study, we chose the IE from bibliographic references aiming at the automatic creation of citation databases. It is possible to extract information from a reference, such as author(s), title, date of publication, etc. Bibliographic references are semi-structured texts with a high degree of variation in their structure [2]. The information to be extracted follows an ordering that, although not rigid, may help the extraction process. To take advantage of this structural ordering, the output delivered by Phase 1 is refined by an HMM.

4.1 Phase 1 - Extraction using a conventional text classifier

As seen above, Phase 1 is divided into three steps:

1. *Fragmentation of the input text*: here, we deployed a set of heuristics based on commas and punctuation.

2. *Feature extraction*: three distinct feature sets were used for describing the fragments: (1) Manual1 (20 features defined in [6]); (2) Manual2 (9 features defined in [3]); and (3) Automatic (100 words directly selected from the training corpus by Information Gain [10]). The first two sets were defined through knowledge engineering and contain features specific to the domain of references such as the occurrence of specific terms (e.g., "journal"), publisher names, etc.

3. *Fragment classification*: we defined 14 different slots for the domain of references: author, title, affiliation, journal, vehicle, month, year, editor, place, publisher, volume, number, pages, and others. We used here three classifiers, implemented using the WEKA environment [9]: the Naive Bayes, the PART (Rules) algorithm and the k-Nearest Neighbour (k-NN).

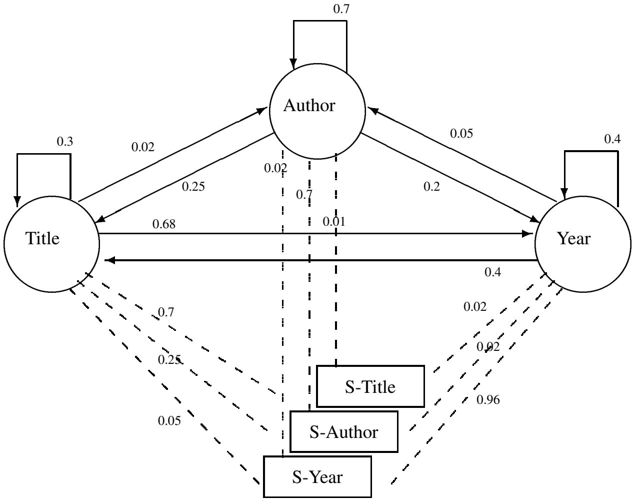

Figure 2. Example of HMM used in Phase 2

1: (S-Author, Author), (S-Journal, Title), (S-Year, Year)
2: (S-Author, Author), (S-Title, Title), (S-Year, Year), (S-Local, Local)
3: (S-Author, Author), (S-Title, Title), (S-Author, Editor), (S-Year, Year)

Figure 3. Sequences for HMM training

4.2 Phase 2 - Refinement of the Results Using an HMM

In this case study, the structure of the HMM was defined as follows: (1) it has one hidden state corresponding to each slot in the output form; and (2) all hidden states were connected to each other. Figure 2 presents a simplified HMM containing 3 symbols, identified by the prefix $S-$, and 3 hidden states, each identified by the name of the associated slot. Figure 3 illustrates training examples of the HMM. In Example 1, for instance, the second fragment was classified in Phase 1 as Journal, but in fact belongs to the Title class.

The transition probability $Pr[s'/s]$ and the emission probability $Pr[t/s]$ are estimated from a set of training sequences by using the following equations defined in [2]:

$$Pr[s'/s] = \frac{Number\ of\ transitions\ from\ s'\ to\ s}{Total\ number\ of\ transitions\ from\ s'} \quad (1)$$

$$Pr[t/s] = \frac{Number\ of\ emissions\ of\ t\ by\ state\ s}{Total\ number\ of\ symbols\ emitted\ by\ state\ s} \quad (2)$$

5 Experiments and Results

The prototype was evaluated using a corpus from a bibliography on computational linguistics[2] which contains 6000 bibliographic references with tags that indicate the class of each text fragment. The collection of references was divided equally into two sets of 3000 references, one for training and the other for testing the system performance.

For each combination of feature set X classifier, we evaluated the performance of our IE system with HMM refinement compared to the system without the HMM. The evaluation measure used was precision, defined as the number of correctly extracted fragments divided by the total number of fragments present in the references.

Table 1 shows the average precision per reference obtained by the system for each combination of feature set and classifier. By comparing the precision obtained with and without the HMM, we verified a performance gain with the use of HMM in all combinations (the gain varied from 3.80 to 22.54 percentage points). The best result was a precision of 81.16%, obtained using the Manual2 set, the classifier kNN and the refinement with the HMM.

The set of features used in Phase 1 strongly influenced system performance. The Automatic set issued the worst average precision rate. However, system performance using this set is clearly improved by the use of the HMM (20.93 percentile points in average), coming closer to the results issued by the other sets. The HMM was able to compensate the use of less expressive feature sets, such as the automatically created sets, thus facilitating the customization of the system to different IE domains.

The system performance also varied depending on the classifier used in Phase 1. However, we observed that the variability of the system performance, considering the classifier used in Phase 1, is lower when the HMM is used.

Table 1. Results obtained in the test corpus

Feature Set	Classifier	Precision without HMM	Precision with HMM	Gain
Manual1	PART	72.17%	76.40%	4.22%
Manual1	Bayes	66.70%	74.72%	8.01%
Manual1	kNN	71.96%	76.28%	4.32%
Manual2	PART	73.48%	77.29%	3.80%
Manual2	Bayes	69.03%	77.27%	8.23%
Manual2	kNN	76.17%	81.16%	4.99%
Automatic	PART	49.91%	72.45%	22.54%
Automatic	Bayes	50.11%	68.25%	18.14%
Automatic	kNN	51.47%	73.57%	22.10%

6 Conclusion

We presented a hybrid machine learning approach for IE in which an HMM is used to refine the initial extraction issued by a text classifier. The hybrid approach was evaluated in the domain of bibliographic references and the performed experiments revealed a significant gain in performance.

One of the main contributions of this work is to have joined two techniques not yet combined in an IE system. The performed experiments showed that the use of an HMM compensated the low performance of less adequate classifiers and feature sets. A high precision average was obtained even with features defined without an expert's effort.

As future work, we highlight the customization of the proposed approach to other domains, the definition of different HMM structures (currently, all hidden states are connected to each other), and the use of machine learning in the generation of the input fragments.

[2] Available in http://liinwww.ira.uka.de/bibliography/Ai/bateman.html

References

[1] D. E. Appelt and D. Israel. Introduction to information extraction technology. In *IJCAI-99 Tutorial*, 1999.

[2] V. Borkar, K. Deshmukh, and S. Sarawagi. Automatic segmentation of text into structured records. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, pages 175–186, 2001.

[3] R. R. Bouckaert. Low level information extraction: a bayesian network based approach. In *TextML*, 2002.

[4] R. Kosala, V. den Bussche, M. Bruynooghe, and H. Blockeel. Information extraction in structured documents using tree automata induction. In *Proc. of the 6th PKDD*, 2002.

[5] N. Kushmerick, E. Johnston, and S. McGuinness. Information extraction by text classification. In *IJCAI-01 Workshop on Adaptive Text Extraction and Mining*, 2001.

[6] C. Nunes and F. A. Barros. Prodext: a knowledge-based wrapper for extraction of technical and scientific production in web pages. In *Proceedings of the International Joint Conference IBERAMIA-SBIA 2000*, pages 106–115, 2000.

[7] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[8] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272, 1999.

[9] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[10] Y. Yang and J. O. Pedersen. A comparative study on feature selection methods in text categorization. In *Procceedings of the 14th ICML*, pages 412–420, 1997.

IEEE COMPUTER SOCIETY