Automatic Information Extraction in Semi-Structured Official Journals

Valmir Macário Filho, Ricardo B. C. Prudêncio, Francisco A. T. De Carvalho
Center of Informatics, Federal University of Pernambuco
Av. Prof. Luiz Freire, s/n 50740-540 Recife/PE BRAZIL
{vmf2,rbcp,fatc}@cin.ufpe.br

Leandro R. Torres, Laerte Rodrigues Júnior
Capital Login
R. da Guia, N 99 50.030-210 Recife/PE BRAZIL
{leandror,laerter}@capitallogin.com.br

Marcos G. Lima
Department of Information Science, Federal University of Pernambuco
Av. dos Reitores, s/n - CEP 50670-901 - Recife/PE BRAZIL
galyndo@gmail.com

Abstract

Information extraction systems are used to extract only relevant text information in digital repositories. The current work proposes an automatic system to extract information in semi-structured official journals. In our approach, given an input document, a Machine Learning (ML) algorithm classifies the document's fragments into class labels which correspond to the data fields to be extracted. The implemented system deployed different features sets and algorithms used in the classification of the fragments. The system was evaluated through experiments on a sample containing 22770 lines of the Pernambuco's Official Journal. The experiments performed revealed, in general, good results in terms of precision, which ranged from 70.14% to 98.63% depending on the feature set and algorithm used in the classification of the fragments.

1 Introduction

A great amount of valuable information is stored in digital repositories of textual documents [1]. A significant part of the information comprised in these repositories is only legible by humans, being hardly manipulated by computer machines. Hence, it is appropriate to develop systems which are capable to automatically extract information on these repositories in order to support specific users' needs [2]. For instance, searching information in historic documents, finding specific sections on a magazine and extracting publications from an official journal.

Official journals are documents that contain publications (e.g., acts, texts of new laws, edicts, decisions) of countries, states, cities and other institutions in the different branches of Executive, Legislative and Judiciary power. Nowadays, these documents are becoming increasingly available in web sites as a new form of information service (e.g., the Official Journal of the European Union[1] and the Official Journal of the Federative Republic of Brazil [2]).

The task of finding specific information of interest in official journals is very difficult due to the great number of publications which are daily available. Although this task can be automated, it is possible to point out some difficulties with regard to this purpose: the lack of rigid models to organize the publications in the documents, no clear delimiters between different publications, the presence of abbreviated words, the presence of orthographic errors, among others.

Documents which present the above-cited characteristics are called semi-structured texts [1]. In order to manipulate such documents, an automatic system called Information Extraction (IE) system may be very suitable. IE systems are able to extract specific information of interest from a repository of textual documents. Each input of an IE system is a textual document and the output is a set of text fragments which correspond to data fields required by the user. The extracted fields can be either directly presented to the user

[1] http://eur-lex.europa.eu
[2] http://portal.in.gov.br/imprensa

or stored in a database for posterior access [3].

The current paper presents an IE system that extracts publications available in official journals. Besides the publications itself, the IE can extract more refined data fields such as the title and subtitle of a publication, the publication body (called process), the notebook of a process and the city of publication. The extracted information is then structured and stored in a database which will be later accessed by the users in appropriate interfaces.

Among the possible approaches to constructing the IE system, we opted to use Machine Learning (ML) algorithms, which are very suitable to deal with semi-structured texts and easily adapted to new domains of application [4]. In the developed system, each official journal given as input is initially fragmented into lines. A ML algorithm classifies each line into pre-defined categories that correspond to the data fields to be extracted. The classification performed by the ML algorithm is based on text features that describe the lines, which include for instance, the presence of specific terms and the matching to regular expressions.

In order to evaluate the system's performance, experiments were performed on a sample of publications from the Official Journal of Pernambuco (Brazil). These publications correspond to the total number of 22,770 lines to be classified. In the experiments, we evaluated the use of different feature sets to described the document lines and three different ML algorithms to classify the lines. For each combination of feature set and algorithm, a 10-fold cross-validation experiment was performed to measure the precision rate obtained over the 22,770 lines. In general, the experiments revealed encouraging results in terms of precision. However, some variation in performance was observed depending on ML algorithm and specially on the evaluated feature set, ranging from 70.14% to 98.63% of precision rate.

Section 2 discusses the topic of Information Extraction. Section 3 describes the proposed system that uses the automatic learning approach. Section 4 presents the experiments and section 5 concludes the article.

2 Information Extraction

The growing availability of information stored in repositories of texts has increased the interest in Information Extraction (IE) [1]. The main objective of an IE system is to recognize pieces of information from texts which correspond to data fields required by the users. IE systems have been used in different contexts to support the automatic construction of databases with the extracted information.

The approach used to construct IE systems strongly depends on the kind of the texts being tackled. Texts can be characterized as non-structured (free text), structured and semi-structured documents [1]. Free texts are written using an unrestricted natural language, and do not fit any regular form or structure. In general, IE systems for free texts use Natural Language Processing techniques involving some kind of semantic and syntactic analysis for the language in which the texts were written [5, 6].

A structured text has a rigid format, commonly produced to be used by a computer. The IE process on these texts can be carried out by using uniform rules in a straightforward manner [7]. Semi-structured texts, in turn, present some regularities in the format and in the order of the desired data fields. However, such texts may present some characteristics which makes it hard to extract information on them. For instance, missing fields, replaced order of fields, lack of delimiters between fields, abbreviate words, among other. Examples of semi-structured texts are bibliographic references, call for papers and official journal documents.

Machine Learning (ML) algorithms have been successfully applied to deal with the task of IE on semi-structured texts [1]. ML algorithms automatically generate extraction rules from labeled corpora, thus favoring a quicker and more efficient customization of IE systems to new domains [4]. Among the ML systems for IE, we cite, for instance, those based on the learning of extraction rules in the form of finite automata and regular expressions [8, 9, 10].

A promissing ML approach applied by different authors is to use learning algorithms as text classifiers for IE [11, 12, 13, 14, 15]. In this approach, the input document is initially broken into fragments which are candidates to fill in the output fields. Next, an ML algorithm classifies each fragment based on its descriptive features (e.g., presence of words, occurrence of numbers, etc). Here, the class values are associated to the required data fields. The extraction is then accomplished by considering the classifications provided by the ML algorithm. The text classification approach has been applied, for instance, to extract information on business cards [11], bibliographic references [12, 13, 15], author affiliations [13], job advertisements [14], among other applications.

The text classification approach for IE is the focus of our work. It has advantages compared to other ML approaches since it may use conventional and easily available algorithms, which is not the case with most IE systems based on automata induction and pattern matching techniques, that use complex and specific learning algorithms [15]. Among the algorithms used as text classifiers, we mention, for instance, bayesian classifiers [13], decision trees [15], Hidden Markov Models [11] and Conditional Random Fields [12].

3 Information Extraction on Official Journal

The current work develops an IE system to extract information from official journals by using ML algorithms. A publication of an official journal is a semi-structured text divided into five main fields: title, sub-title, notebook, city

and process. Figure 1 shows an example of data fields extracted from an official journal. Correctly extracting these fields is a challenge due to the complexity associated to the official journal construction.

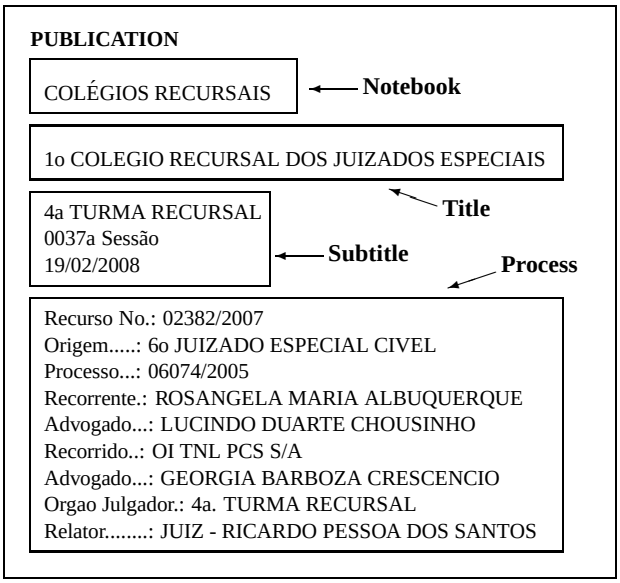

Figure 1. Example of data fields extracted from a publication.

Some difficulties to extract information from official journals can be mentioned here: (1) fields may present very similar patterns (e.g., the sentence "Edital de Intimação", may appear in the beginning of both fields subtitle and process); (2) absent fields (e.g., the field city of the example illustrated in Figure 1 was omitted); and (3) presence of abbreviated patterns (e.g., the word "Process" is in many publications abbreviated to "Proc.").

Figure 2 shows a generic architecture of the IE system which deployed the text classification approach for IE. This architecture has three steps:

1. *Fragmentation*: the input text is broken into fragments which are the candidates for filling in the required data fields. In our domain, the fragments correspond to the text lines.

2. *Feature extraction*: a vector of features is created to describe each text fragment and it is used in the classification of the fragment.

3. *Fragment classification*: a learned classifier associates each input fragment to a class label associated to a data field.

The above process automatically provides labels for the fragments (lines) of the input document (official journal). Each extracted data field will be composed by one or more lines, taking into account the labels provided by the ML classifier. The steps of our system will be detailed in the next subsections.

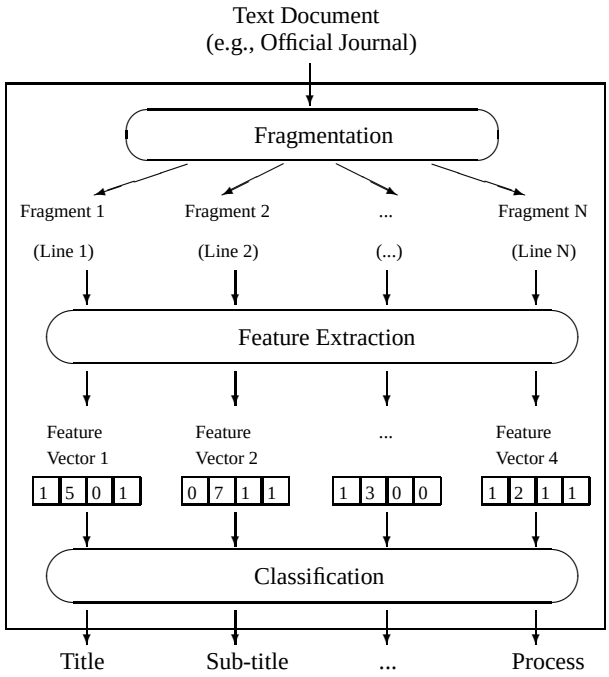

Figure 2. IE system architecture.

3.1 Fragmentation

In this step, the input document is fragmented into short pieces to be later associated to data fields. In general, the documents may be fragmented based on text delimiters, which may be punctuation marks, white spaces, end-of-line characters, paragraph characters, among others.

In some IE tasks being tackled in previous work, it is necessary to deploy different delimiters in order to generate text fragments which are good candidates to be associated to the required data fields (e.g., [4, 11, 15]). In our domain of application, however, the data fields to be extracted always correspond to one or more complete lines of text, and hence, the delimiter used to fragment the input documents was solely the end-of-line character.

3.2 Feature Extraction

In this step, the features used to classify the fragments are defined. This task was accomplished by considering a domain vocabulary, regular expressions and text formatting

features. More specifically, in this work, we evaluated three different sets of features:

1. *Vocabulary*: it corresponds to 180 words which were considered as relevant words to distinguish the different data fields. Each feature, in this case, is a boolean variable which indicates whether a specific word is observed in the text fragment. The *Vocabulary* feature set required effort of a domain expert to be defined.

2. *Regular Expression*: the data fields in our domain may match, in some cases, to patterns that can be represented as regular expressions. Hence, in this feature set, a domain expert defined regular expressions for each main field to be extracted. Each feature in this case is a boolean variable which indicates whether the fragment matches to a specific regular expression. A total number of 280 regular expressions was defined for this feature set.

3. *General*: this set, composed by 15 features, was defined based on the Bouckaert's work [13]. It considers text formatting features which are more generally applicable than the features considered in the *Vocabulary* and the *Regular Expression* sets. It includes, for instance, a feature indicating whether the line starts with an uppercase letter, a feature indicating whether it contains numbers or enumerations, among other general characteristics.

We highlight here that the above sets are composed by local descriptive features which do not consider the order of the fragments. Hence, these sets may be not adequate to identify sequential dependencies in the data fields that could be eventually useful to classify the document fragments. For instance, a fragment should not be classified as title if the previous fragment is classified as subtitle. The order of the fragments is useful to perform a globally optimal classification of the whole sequence of input fragments.

In order to take into account sequential dependences between data fields, we evaluated to represent each fragment not solely based on its own features, but also considering the features of adjacent fragments. More specifically, each fragment is described by its own local features plus the features of its just precedent and following fragments. This representation approach is referred in the literature as the *Sliding Window* (SW) approach [16], which was used for instance in [13] in the IE on bibliographic references and on author affiliations.

3.3 Fragment Classification

In this step, a machine learning classifier receives the sequence of feature vectors describing the input text fragments and then returns the sequence of the class labels associated to the fragments. The classifications are performed based on knowledge acquired in a learning process which considers a set of manually labeled fragments. Each learning example stores the feature vector describing a fragment and the class label correctly associated to the fragment.

In our system, there are ten possible class labels: *iprocess*, *process*, *ititle*, *title*, *isubtitle*, *subtitle*, *notebook*, *city*, *nil* and *blank*. The class labels *iprocess* and *process* are associated to the lines which correspond to the process field (the prefix *'i'* is used to designate the first line of the field). We opted to use two labels for this data field since it can be composed by more than one line and specific patterns (expressed as regular expressions) only matches with the first line of the field. Similarly, the labels *ititle* and *title* correspond to the publication's title, and the labels *isubtitle* and *subtitle* represent the publication's sub-titles. The labels *notebook* and *city* are respectively associated to the data fields notebook and city of the publication. The label *nil* is used to indicate lines which do not correspond to any data field. The label *blank* matches to a black line, and it is important in the extraction process since a each individual data field has one or more blank lines before.

Finally, the extraction of the data fields is performed by deploying a set of rules which take into account the classification provided by the ML algorithm for the document lines. For instance, each portion of the input document between two lines classified as *iprocess* is extracted as an individual publication. As another example of rule, the body of the publication will be composed by the concatenation of adjacent lines classified as *process*. Similarly, the title of the publication will be composed by concatenating the lines classified as *title*.

In this step, we evaluated the use of three classifiers, each one representing a different family of learning algorithms: (1) the PART algorithm for inducing decision rules [17], (2) the Naive Bayes classifier [18] and the Support Vector Machine (SVM) classifier [19]. Next section, we present the experiments that evaluated the performance of these classifiers in a corpus of publications.

4 Experiments and Results

In this section, we present the description of the experiments performed to evaluate the implemented IE system, as well as the results obtained on a corpus of publications.

4.1 Experiments Description

In our work, the performed experiments were based on a corpus of publications collected from the Judiciary segment of the Official Journal published by the State of Pernambuco, Brazil. The sample corresponds to the Official Jour-

nal pages published from 8 to 14 February (a week of publication). The total number of lines in the corpus is 22,770 (approximately, 4,000 lines a day). Each line was correctly labeled by an expert into one of the 10 possible class labels.

The performance of the IE system was evaluated for 21 different scenarios (i.e., different combinations of feature sets *versus* classifiers). As seen in Section 3.2, we used 3 classifiers: the PART, the Naive Bayes and the SVM. The SVM algorithm uses the radial kernel function. The feature sets presented in section 3.1 were combined in 7 different ways: (1) General (15 features); (2) Vocabulary (180 features); (3) Regular Expression (RE) (280 features); (4) General + Vocabulary (195 features); (5) General + RE (295 features); (6) Vocabulary + RE (460 features); and (7) General + Vocabulary + RE (475 attributes).

The same above scenarios were also applied to evaluate the usefulness of the Sliding Window (SW) approach. As said in section 3.2, in this approach the text classifier receives as input the features of the fragment being classified and the features of the adjacent fragments. This battery of experiments aims to evaluate the need for considering sequential information in the extraction process. The window has overlapping and the size used on this work is 3.

For each different scenario of feature set *versus* classifier, with and without SW, we applied a 10-fold cross-validation procedure to evaluate system's performance. The evaluation measure used was precision, defined as the number of correctly classified lines divided by the total number of lines present in the corpus of experiments.

4.2 Results

Table 1 shows the precision obtained in the experiments. The best observed result was 98.63%, obtained by using the PART algorithm, the combination of all feature sets and the SW representation. The worst result was 70.14% of precision obtained by using the SVM algorithm, the RE feature set, without SW. By comparing the experiments with and without SW, we observed an improvement in precision on 19 of the 21 combinations of feature set and classifier. This result indicates that the use of sequential information can improve the performance of the text classifiers.

Table 2 presents the average precision obtained for each classifier. The best average result was obtained by using the PART algorithm, considering both the experiments without SW (87.25%) and with SW (88.97%). However, the performance obtained by using the PART algorithm was not so different from the performance obtained by using the other classifiers. For all evaluated classifiers, we observed that the use of SW improved the average precision rate.

In Table 3, we can see the average precision obtained in the experiments for each evaluated feature set. The best average results (from 94.47% to 95.85%) were obtained in the scenarios in which the feature set General was used. In fact, by using solely the feature sets Vocabulary and RE, the performance of the IE system was strongly harmed. On 6 of the 7 combinations of feature sets, the use of SW improved the performance of the IE system.

Table 1. Results obtained in the 10-fold cross validation experiment on a set of 22,770 lines.

Feature Set	Classifier	Precision without SW	Precision with SW	Difference
General	PART	96.44	98.29	1.85
	Bayes	91.57	91.04	-0.53
	SVM	95.41	97.18	1.77
Vocabulary	PART	75.28	76.43	1.15
	Bayes	73.58	81.64	8.06
	SVM	73.09	73.14	0.04
RE	PART	73.09	75.98	2.88
	Bayes	73.09	76.04	2.95
	SVM	70.14	75.54	5.40
General +Vocabulary	PART	97.12	98.57	1.45
	Bayes	92.46	92.96	0.49
	SVM	94.01	95.57	1.56
General +RE	PART	96.44	98.36	1.93
	Bayes	92.26	91.30	-0.95
	SVM	95.62	97.09	1.47
Vocabulary +RE	PART	75.28	76.57	1.29
	Bayes	73.56	83.00	9.44
	SVM	73.09	75.32	2.22
General +Vocabulary+RE	PART	97.13	98.63	1.50
	Bayes	92.46	93.17	0.71
	SVM	93.89	95.76	1.88

Table 2. Average precision for each classifier.

Classifier	Average Precision without SW	Average Precision with SW
PART	87.25	88.97
Bayes	84.14	87.02
SVM	85.03	87.08

5 Conclusion

The current work proposes an automatic IE system to extract information in semi-structured official journals. The implemented system deployed the text classification approach for IE, which revealed to be adequate for our purpose. We highlight that the application of text classifiers for IE in the domain of Official Journals is an original work.

In our experiments, we evaluated different features sets and learning algorithms in the classification of the text fragments. We observed a precision in the classification task which specially depended on the used feature set. We

Table 3. Average precision for each feature set.

Feature set	Average Precision without SW	Average Precision with SW
General	94.47	95.50
Vocabulary	73.98	77.07
RE	72.11	75.85
General+Vocabulary	94.53	94.27
General+RE	94.77	95.58
Vocabulary +RE	73.98	78.30
General+Vocabulary +RE	94.49	95.85

also observed that an improvement in performance can be yielded when sequential information of the fragments is taken into account.

The IE system can be extended to other domains of application. Additionally, as future work, other approaches can be used to construct the feature sets. We also intend to use evaluate sequential learning algorithms, such as Hidden Markov Models and Conditional Random Fields to classify the fragments.

Acknowledgments: The authors would like to thank CNPq (Brazilian Agencies), Capital Login and Aileader Technologies for their financial and structure support.

References

[1] Turmo, J., Ageno, A. and Català, N. Adaptive Information Extraction. *ACM Computing Surveys* 38(2):4 2006.

[2] Eikvil, L. Information extraction from the world wide web: a survey. *Technical Report 945*, Norwegian Computing Center, 1999.

[3] Appelt, D. and Israel, D. Introduction to Information Extraction Technology. *IJCAI-99 Tutorial*, Stockholm, Sweden, 1999.

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

[5] Hobbs, J. R., Appelt, D., Israel, D. and Kameyama, M. Fastus: A cascaded finite state transducer for information extraction from natural language text. In *Finite State Devices for Natural Language Processing*, University of Amsterdam, 1997.

[6] Soderland, S., Fisher, D., Aseltine, J. and Lehnert, W. Crystal: Inducing a conceptual dictionary. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1314-1319, 1995.

[7] Freitag, D. Information extraction from HTML: Application of a general machine learning approach. In *National Conference on Artificial Intelligence*, pp. 517-523, 1998.

[8] Kosala, R., Van Den Bussche, J., Bruynooghe, M. and Blockeel, H. Information extraction in structured documents using tree automata induction. *Lecture Notes in Computer Science* 2431:299-310, 2002.

[9] Muslea, I., Minton, S. and Knoblock, C. Active learning with multiple views. *Journal of Artificial Intelligence Research* 27:203-233, 2006.

[10] Feldman, R., Rosenfeld, B. and Fresko, M. TEG: a hybrid approach to information extraction. *Knowledge and Information Systems* 9(1):1-18, 2006.

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

[12] Laferty, J., McCallum, A. and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*. 2001.

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

[14] De Sitter, A. and Daelemans, W. Information extraction via double classification. In *Proceedings of International Workshop on Adaptive Text Extraction and Mining*, pp. 66-73, 2003.

[15] Silva, E. F. A., Barros, F. A. and Prudêncio, R. B. C. A Hybrid Machine Learning Approach for Information Extraction. In *Proceedings of the 6th International Conference on Hybrid Intelligent Systems*, Auckland, New Zealand, 2006.

[16] Dietterich, T. G. Machine learning for sequential data: a review. *Lecture Notes in Computer Science* 2396:15-30, 2002.

[17] Frank, E. and Witten, I. H. Generating accurate rule sets without global optimization. In *Proceedings of the 15th International Conference on Machine Learning*, pp. 144-151, 1998.

[18] John, G. H. and Langley, P. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pp. 338-345, 1995.

[19] Vapnik, V. N. *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.