A Machine Learning Approach to Information Extraction

Alberto Téllez-Valero[1], **Manuel Montes-y-Gómez**[1,2], **Luis Villaseñor-Pineda**[1]

[1] Language Technologies Group, Computer Science Department,
National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico.
{albertotellezv, mmontesg, villasen}@inaoep.mx
[2] Department of Information Systems and Computation,
Polytechnic University of Valencia, Spain.
{mmontes}@dsic.upv.es

Abstract. Information extraction is concerned with applying natural language processing to automatically extract the essential details from text documents. A great disadvantage of current approaches is their intrinsic dependence to the application domain and the target language. Several machine learning techniques have been applied in order to facilitate the portability of the information extraction systems. This paper describes a general method for building an information extraction system using regular expressions along with supervised learning algorithms. In this method, the extraction decisions are lead by a set of classifiers instead of sophisticated linguistic analyses. The paper also shows a system called *TOPO* that allows to extract the information related with natural disasters from newspaper articles in Spanish language. Experimental results of this system indicate that the proposed method can be a practical solution for building information extraction systems reaching an F-measure as high as 72%.

1 Introduction

The technological advances have brought us the possibility to access large amounts of textual information, either in the Internet or in specialized collections. However, people cannot read and digest this information any faster than before. In order to make it useful, it is often required to put this information in some sort of structured format, for example, in a relational database.

The information extraction (IE) technology is concerned with structuring the relevant information from a text of a given domain. In other words, the goal of an IE system is to find and link the relevant information while ignoring the extraneous and irrelevant one [2]. The research and development in IE have been mainly motivated by the Message Understanding Conferences (MUC[1]). These conferences provide a decade of experience in the definition, design, and evaluation of this task.

According to the MUC community, the generic IE system is a pipeline of components, ranging from preprocessing modules and filters, to linguistic components for syntactic and semantic analysis, and to post-processing modules that construct a final

[1] www.itl.nist.gov/iaui/894.02/related_projects/

answer [4]. These systems deal with every sentence in the text and try to come up with a full-scale syntactic, semantic, and pragmatic representation. Evidently, they have serious portability limitations since their construction demands a lot of hand-crafted engineering to build the required grammars and knowledge bases.

On the other hand, empiric or corpus based methods are encouraging for the development of IE systems, and in general for many computational linguistics tasks (see [7] for a study). These methods automate the acquisition of knowledge by means of training on an appropriate collection of previously labeled documents. Unlike the traditional approach, they are based on pattern recognition instead of language understanding, and use shallow knowledge instead of deep knowledge. Their main advantages are portability and robustness.

Most current IE systems apply linguistic techniques for text pre-processing and use empiric methods to automatically discover morpho-syntactic extraction rules. This combined scheme produces satisfactory results even when the common errors at the pre-processing stage impose a barrier at the output accuracy. It facilitates the domain portability, but complicates the extensive usage of the IE technologies in other languages than English that lack of robust natural language processing resources.

In this paper we propose a general empiric method for building IE systems. This method avoids using any kind of sophisticated linguistic analysis of texts. It models the IE task as a text classification problem [13]. Basically, it is supported on the hypothesis that the lexical items around the interesting information are enough to learn most extraction patterns. Therefore, the main characteristic of this proposal is its small dependence to the target language.

In order to evaluate this method, we present a system called *TOPO*. This system allows to extract information about natural disasters from news reports in Spanish language. Our results demonstrate that our approximation can be fruitfully used to extract information from free-text documents.

The rest of the paper is organized as follows. Section 2 describes previous work on information extraction using machine learning techniques. Section 3 presents our approach to information extraction based on text classification methods. Section 4 shows a general IE system architecture based on this approach. Section 5 describes a real-world application and shows the results. Finally, section 5 concludes the discussion.

2 Related Work

The use of machine learning (ML) methods in IE applications is mainly focused on the automatic acquisition of the extraction patterns. These patterns are used to extract the information relevant to a particular task from each single document of a given collection (see [9,10,17] for a survey). Current IE approaches, supported on supervised ML techniques, are divided in the following three categories:

Rule Learning. This approach is based on a symbolic inductive learning process. The extraction patterns represent the training examples in terms of attributes and relations between textual elements. Some IE systems use propositional learning (i.e. zero order logic), for instance, AutoSlog-TS [11] and CRYSTAL [15], while others perform a relational learning (i.e. first order logic), for instance WHISK [16] and

SRV [3]. This approach has been used to learn from structured, semi-structured and free-text documents.

Our method is related to the SRV system in that it models the IE task as a classification problem. However, it applies Inductive Logic Programming and uses information about negative examples.

Linear Separators. In this approach the classifiers are learned as sparse networks of linear functions (i.e. linear separators of positive and negative examples). It has been commonly used to extract information from semi-structured documents (see for instance SnoW-IE [12]). It has been applied in problems such as: affiliation identification and citation parsing [1], extraction of data from job ads [18], and detection of an e-mail address change [5].

In general, the IE systems based on this approach present an architecture supported on the hypothesis that looking at the words combinations around the interesting information is enough to learn the required extraction patterns. Their main advantage is that a deep linguistic analysis is not necessary; instead classification techniques are used to find the desired information.

Our method is similar to all these systems. It is based on the same hypothesis. However, it is suited for extracting more general and diverse kinds of information. In some degree our research attempts to empirically determine the limits of this approach when dealing with a complex domain and free texts instead of semi-strcutured documents.

Statistical Learning. This approach is focused on learning Hidden Markov Models (HMMs) as useful knowledge to extract relevant fragments from documents. For instance, [14] presents a method for learning model structure from data in order to extract a set of fields from semi-structured texts. This method is similar to ours in that it considers just the lexical information of texts .

3 Information Extraction as a Classification Problem

Our IE method, like the linear separator approach, is supported on the idea that looking at the words combinations around the interesting information (i.e. the context) is enough to learn the required extraction patterns. Therefore, this method considers two main tasks:

1. Detect all the text segments having some possibility to be part of the output template.
2. Select, from the set of candidate text segments, those that are useful to fill the extraction template.

The figure 1 illustrates this process with a simple example about a hurricane news report. The following subsections describe the purpose and techniques used on each task.

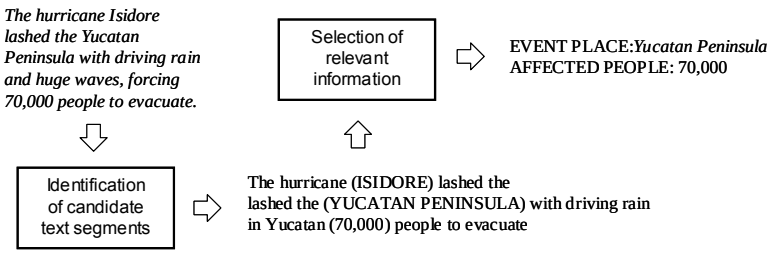

Figure 1. Information extraction as text classification

3.1 Detecting candidate text segments

The goal of this task is to detect the majority, if not all, of the text segments having some possibility to take place in the extraction template. Since most IE applications consider just the extraction of simple factual data, our method is focused on detecting the text segments expressing names, quantities and temporal data.

In order to identify the candidate text segments we use a *regular expression analysis*. This kind of analysis is general and robust, produces high levels of recall, and is consistent with our purpose of using the less as possible of linguistic resources.

The first part of the figure 1 shows this task. The uppercase words correspond to the candidate text segments of the input text. For each candidate text segment its context (the k neighbor words from left and right) is also extracted.

3.2 Selecting the relevant information

The goal of this task is to capture the text segments that must be part of the output template, in other words, it is responsible to classify the text segments into relevant and irrelevant (i.e. to extract or not).

The classification is based on *supervised learning techniques*. In this framework, each candidate text segment is classified according to its lexical context.

In contrast to the previous task, the selection of the relevant information must achieve a high precision rather than a high recall. This situation motivates us to use a pool of learning methods in order to specialize a different classifier for each type of output data. For instance, build a classifier for names, other for dates and another for quantities.

The second part of the figure 1 illustrates this task. There, the classifier uses the contextual information to discard the text segment (ISIDORE) as not relevant to the output template, and also to define (YUCATAN PENINSULA) and (70,000) as the dissaster place and the number of affected people respectivaly.

4 A general IE system architecture

This section describes a general IE system achitecture based on our approach of "information extraction by text classification" (refer to the section 3). This architecture is shown in the figure 2. It consists of two basic stages: text filtering and information extraction.

It is important to notice that both stages are fully suppported on supervised machine learning algorithms. Moreover, both stages are trainned with the same corpus, and both considers just the lexical information for training.

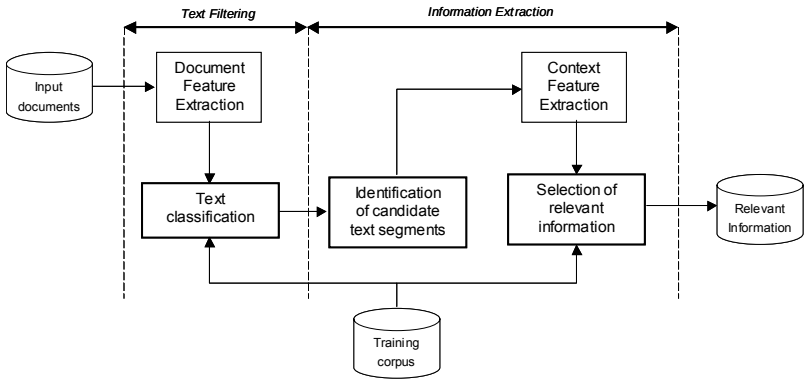

Figure 2. General IE system architecture

The main characteristic of this architecture is its portability. It is language independent since the training features and the candidate text segments are selected and identified basen on simple lexical patterns and criteria. Also, it can be easily adapted to different domain applications by constructing a small training corpus. Our experiments, refer to the following section, indicates that some hundreds of training examples are enough to reach an acceptable level of output accurancy.

5 A Case Study: Natural Disasters Reports

In this section we introduce the system *TOPO*. A system that extracts information related with natural disasters from newspaper articles in Spanish language. This system was inspired by the work carried out by the network of Social Studios in Disasters Prevention in Latin America [6].

TOPO allows to extract the following information: (i) information related with the disaster itself, i.e. it date, place and magnitude; (ii) information related with the people, for instance, number of dead, wounded, missing, damaged and affected persons; (iii) information related with the buildings, e.g. number of destroyed and affected houses; and (iv) information related with the infrastructure, that is, number of affected hectares, economic lost, among others.

Currently, *TOPO* works with news reports about: hurricanes, forest fires, inundations, droughts, and earthquakes. The following subsections present its technical characteristics and some experimental results.

5.1 Technical Characteristics

Document feature extraction. The documents are represented as boolean vectors indicating the presence or absence of certain words in the texts. The Information Gain technique was used to avoid a high dimensionality of the feature space. The result was a vector formed by 648 terms obtained from a vocabulary of 26,501 terms from a collection of 534 news reports.

Text classification. We experimented with four different machine learning algorithms [8]: Support Vector Machines (SVM), Naive Bayes (NB), C4.5 and k-Nearest Neighbors (kNN). This selection was based on recent studies that define these classifiers as the best ones for text processing tasks [13].

Candidate text selection. In order to identify the candidate text segments (i.e., names, dates and quantities) from Spanish texts we use the following grammar:

```
Entity_name      → name |
                   Name connect_name entity_name
Entity_date      → month |
                   month connect_date number |
                   number connect_date entity_date
Entity_quantity  → number(. number)? |
                   number(. number)? entity_quantity
```

In this grammar, the terminals symbols generate groups of chains given by the following regular definitions:

```
name          → [A-Z][A-Za-z]*
connect_name  → de | la | ... | ϵ
month         → enero | ... | diciembre
connect_date  → de | - | ... | ϵ
number        → [0-9]+
```

In addition, we are using a dictionary of names and numbers to treat some grammar exceptions (e.g.: to identify textual quantity expressions and to eliminate words starting with a capital letter but expressing a not valid named entity).

Context feature extraction. This process represent the context of the candidate text segments as a vector of nominal attributes, i.e. the words surrounding the text segments.

In the experiments, we consider context sizes from 1 to 14 terms. In addition, we evaluate several ways of defining this context: (i) using the original surrounding words; (ii) not using stop words as attributes; (iii) using the root of the words; and (iv) using entity tags, i.e., substituting candidate text segments in the context for a tag of name, date or quantity.

Selection of relevant information. It considered the same classifiers used on the text classification task. However, as said elsewhere above, we attempt to specialize each classifier in a different type of output data (i.e., one for the names, other for the dates and another one for the quantities).

5.2 Experimental Results

Text filteirng stage. It was evaluated on a test set of 134 news reports. The evaluation considered the metrics of precision, recall and F-measure[2] adapted to the text classification task [13].

Table 1 resumes the best results we obtained using the SVM algorithm. It is important to mention that these results are equivalent to those reported for similar domains. For instance [13] reports an F-measure from 72% to 88% on the classification of news reports from the Reuters collection.

Table. 1. Results for the text filtering task

Disaster	Precision	Recall	F-measure
Forest fire	100	96	98
Hurricane	93	87	90
Inundation	82	93	88
Drought	86	60	71
Earthquake	92	100	96

Information extraction stage. This stage was evaluated on a training set of 1353 text segments –that represent the context of names, dates, and quantities– taken randomly from 365 news reports about natural disasters. Just the 55% of the training examples represent relevant information to be extracted.

In order to evaluate the performance of the information extraction task, we used the precision, recall, and F-measure metrics as defined by the MUC community.

$$precision = \frac{Number\_correct}{Number\_correct + Number\_incorrect + Number\_spurious} \quad (1)$$

$$recall = \frac{Number\_correct}{Number\_correct + Number\_incorrect + Number\_mis\sin g} \quad (2)$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

The table 2 resumes the experimental results. This outcome correspond to a context of size eight (i.e., four words to the left and four words to the right) for names and dates text segments, and a context of size six (i.e., three words to the left and three words to the right) for the quantities text segments. The best classifiers were SVM for names and quantities, and kNN for dates.

In general, we obtained a 72% average F-measure for the information extraction task. The precision measures were greater than the recall ones. This indicates that our system is more accurate than complete. We think this situation can be compensated with the redundancies existing in the news reports.

[2] Precision is the proportion of documents placed in the category that are really in the category, and recall is the proportion of documents in the category that are actually placed in the category. The F-measure is a lineal combination of both proportions.

Table. 2. Results for the information extraction task

Information	Precision	Recall	F-measure
Disaster date	95	95	95
Disaster place	42	81	55
Disaster magnitude	75	89	82
People dead	65	91	76
People wounded	89	86	88
People missing	79	73	76
People damaged	72	64	68
People affected	50	51	50
Houses destroyed	59	82	69
Houses affected	63	37	47
Hectares affected	66	96	78
Economic lost	80	76	78

These results are equivalent to those reported for similar IE applications. For instance, at MUC-6, where were analyzed news about managerial successions, the participants obtains F-measures lower than 94% for the entity recognition task and measures lower than 80% for the template filling (information extration task).

Finally, it is important to mention that *TOPO* is currently being used for automatically populating a database of natural disasters from Mexican news reports. The system was implemented in Java using the Weka open source software.

6 Conclusions

This paper presents a general approach for building an IE system. This approach is supported on the idea that looking at the word combinations around the relevant text segments is sufficient enough to learn to discriminate between relevant and irrelevant information.

In the proposed approach the information extraction is done by a combination of regular expressions and text classifiers. The use of these methods allows to easily adapt an IE application to a new domain. In addition, it avoids the employment of any kind of sophisticated linguistic recourse, which defines this approach as language independent.

Our experiments demonstrated the potential of this approach. Using a very small training set we reached an average F-meausre of 72% for the extraction task.

The main disadvantages of the proposed approach are: on the one hand, that it is not possible to extract the information expressed in an implicit way. On the other hand, that it is complicated to extract and link the information from documents reporting more than one interesting event. We believe that these problems can be partially solved using some level of linguistic analysis as a preprocessing stage, just before applying the regular expression analysis.

Acknowledgements

We would like to thank CONACyT for partially supporting these work under grants 171610, 43990A-1 and U39957-Y, and to the Secretaría de Estado de Educación y Universidades de España.

References

1. Bouckaert, R.: Low level information extraction. In Proceedings of the workshop on Text Learning (TextML-2002), Sydney, Auxtralia (2002)
2. Cowie, J., Lehnert, W.: Information Extraction. Communications of the ACM, Vol. 39, No. 1 (1996) 80-91
3. Freitag, D.: Machine Learning for Information Extraction in Informal Domains. Ph.d. thesis, Computer Science Department, Carnegie Mellon University, (1998)
4. Hobbs, J. R.: The Generic Information Extraction System. In proceedings of the Fifth Message Understanding Conference (1993)
5. Kushmerick, N., Johnston, E., McGuinness, S.: Information Extraction by Text Classification. Seventeenth International Join Conference on Artificial Intelligence (IJCAI-2001), N. Kushmerick Ed. Adaptive Text Extraction and Mining (Working Notes), Seattle, Washington (2001) 44-50
6. LA RED: Guía Metodológica de DesInventar. OSSO/ITDG, Lima (2003)
7. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
8. Michel, T.: Machine Learning. McGraw Hill, (1997)
9. Muslea, I.: Extraction Patterns for Information Extractions Tasks: A Survey. In Proceedings of the AAAI Workshop on Machine Learning for Information Extraction (1999)
10. Peng, F.: Models Development in IE Tasks - A survey. CS685 (Intelligent Computer Interface) course project, Computer Science Department, University of Waterloo (1999)
11. Riloff. E.: Automatically Generating Extraction Patterns from untagged text. In proceedings of the 13th National Conference on Artificial Intelligence (AAAI), (1996) 1044-1049
12. Roth, D., Yih, W.: Relational Learning Via Propositional Algorithms: An Information Extraction Case Study. In Proceedings of the 15th International Conference on Artificial Intelligence (IJCAI), (2001)
13. Sebastiani, F.: Machine Learning in Automated Text Categorization: a Survey. Technical Report IEI-B4-31-1999, Istituto di Elaborazione dell'Informazione (1999)
14. Seymore, K., McCallum, A., Rosenfeld, R.: Learning Hidden Markov Model structure for Information Extraction. In Proceedings of the 20th National Conference on Artificial Intelligence (AAAI), (1999).
15. Sonderland, S., Fisher, D., Aseltine, J., Lehnert, W.: CRYSTAL: Inducing a Conceptual Dictionary. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), (1995) 1314-1321
16. Sonderland, S.: Learning Information Extraction Rules for Semi-Structured and Free Text. Machine Learning, No. 34 (1999) 233-272
17. Turno, J.: Information Extraction, Multilinguality and Portability. Revista Iberoamericana de Inteligencia Artificial. No. 22 (2003) 57-78
18. Zavrel, J., Berck, P., Lavrijssen, W.: Information Extraction by Text Classification: Corpus Mining for Features. In Proceedings of the workshop Information Extraction meets Corpus Linguistics, Athens, Greece (2000).