Classification Based Automatic Information Extraction System from Free Text

MYAT MYO NWE WAI,
Lecture, University of Computer Studies,
Banmaw, Myanmar
wmyonwesit@gmail.com

Abstract

The ever increasing on-line text information can be made available to automatic processing by information system. Several machine learning techniques have been applied in order to build automatic information that rivaling knowledge engineering approach. The past reporting information extraction system using machine learning techniques that expend the separate classifiers for each category of entities. In this paper, we introduce the new classification based information extraction system utilizing only one Random Forest classifier for all candidate entities to save the computational costs of algorithms. Our approach extends the original idea of Random Forest to deal with the data sparseness problem in information extraction engine. Experimental results of this system indicate that the proposed method can be a practical solution for building extraction system reaching an F-measure as high as 87.5%.

1. Introduction

As more and more text becomes available on line there is a growing need for system that extracts information automatically from text data. Information extraction systems have been developed for writing styles ranging from structured text with tabular information to free text such as news stories. Our system is deal with extracting information from free text. When they are applied to unstructured texts, data sparseness becomes a problem. Data sparseness is relevant for (i) the size of the training data: the more sparse data are the more example are needed for training, (ii) quality of results, sparse data cause the generated rule to be applicable to a limited number of cases, overfitting the training examples and therefore, affecting effectiveness on unseen cases.

Information extraction can be done manually by having an expert user created rules that will extract the desired entities. This is difficult, expensive and time consuming process. We are interested in Automated Information Extraction system. The annotated documents are used as examples for a Machine Learning algorithm. A human annotator annotates examples of the entities that we want to extract. The use of machine learning methods in IE applications is mainly focused on the automatic acquisition of the extraction patterns. Thus the role of the human is reduced to labeling example entities in documents rather than having to construct complex sets of rules. The complexity of identifying the best rules is left to the learning algorithm. There are two Machine Learning approaches to build the information extraction system: pattern based extraction system and classification based extraction system. In former approach, the extraction process is very much its template matching task between the extraction patterns and the sentences. The later approach allows an extremely large number of features arise to information extraction in learning text analysis stage. Thus, a classification model will become necessary to solve the data sparseness problem of IE.

To extract information from text documents, most IE systems rely on a set of extraction patterns. Each extraction pattern is defined based on the syntactic and/ or semantic constraints on the positions of desired entities within natural language sentences. The IE systems also provide a set of pattern templates that determine the kind of syntactic and semantic constraints to be considered. In this paper, we argue that such pattern templates restrict the kind of extraction patterns, we first propose to model the content and context information of a candidate entity to be extracted as a set of features. In particular we use Random Forest classifier based on these feature sets to build for automatic information extraction system.

The rest of the paper is organized as follow. Section 2 discusses the related works of the other information extraction system. Section 3 and 4 present the random forest classifier and our basic approach that treats IE as a classification task. Section 5 describes a general overview of how proposed Information Extraction System works with a detailed explanation of the architecture of the system and with experimental results of our extraction engine for Management Succession domain.
and conclusion and future work of the paper follows in Section 6.

2. Related Work

Machine Learning (ML) method used in Information Extraction applications is mainly focused on the automatic acquisition of the extraction rules. We reviewed on the three supervised ML techniques: rule learning, linear separator approach and statistical learning approach. Rule learning is based on a symbolic inductive learning process. The extraction patterns represent the training examples in terms of attributes and relations between textual elements. AutoSlog_TS [10] and CRYSTAL [15] systems use propositional learning. AutoSlog_TS [10] designed to obviate the need for special training data. CRYSTAL [15] is another system that inductive generates a dictionary of concept definitions that cover the positive examples contained in the training texts.

WHISK [14] and SRV [4] systems performs a relational learning. WHISK [14] use covering algorithm, including a set of rules from hand-tagged training examples. SRV [4] generates first-order logic extraction pattern that are based on attribute-value tests and the relational structure of the documents. Our method used in information extraction engine is related to the CRYSTAL [15] system that it applies covering algorithm models to induce the extraction patterns. However our system models the IE task as a classification problem from free text.

In linear separator approach the classifiers are learned as sparse networks of linear functions. Snow-IE [12] has been commonly used to extract information from semi-structured document. It has been applied in problem such as: affiliation identification and citation parsing, extraction of data from job advertisement, and detection of and email address change.

Statistical learning approach is focused on learning Hidden Markov Model (HMMs) [13] as useful knowledge to extract relevant fragments from documents. K. Seymour et al [13] present a method for learning model structure from data in order to extract a set of fields from semi-structured texts.

3. Operations of Random Forest Classifier

Random Forest induction (Beriman, 2001) is an ensemble method. An ensemble learning algorithm constructs a set of classifiers, then classifies new data points by taking a vote of the predictions of each classifier. A necessary and sufficient condition for an ensemble of classifier to be more accurate than any of its individual members is that the classifiers are accurate and diverse. Two classifiers are diverse if they make different errors on new data points. Random forests try to increase diversity among the classifiers by re-sampling the data, and by changing the feature sets over the different tree induction processes. The exact procedure is as follows:

(i) for \( i = 1 \) to \( k \) do

- build dataset \( D_i \) by sampling with replacement from dataset \( D \)
- learn a decision tree \( T_i \) from \( D_i \) using randomly restricted feature sets

(ii) make predictions according to the majority vote of the set of \( k \) trees.

Random forest uses the gini index from the CART learning system to construct decision trees. The gini index of node impurity is the measure most commonly chosen for classification type problems. If a data set \( T \) contains examples from \( n \) classes, gini index, \( \text{Gini}(T) \) is defined as

\[
\text{Gini}(T) = 1 - \sum_{j=1}^{n} P_j^2
\]

where \( P_j \) is the relative frequency of class \( j \) in \( T \). If a dataset \( T \) is split into two subsets \( T_1 \) and \( T_2 \) with size \( N_1 \) and \( N_2 \) respectively, the gini index of the split data contains examples from \( n \) classes, the gini index \( \text{Gini}_{\text{split}}(T) \) is defined as

\[
\text{Gini}_{\text{split}}(T) = \frac{N_1}{N} \text{gini}(T_1) + \frac{N_2}{N} \text{gini}(T_2)
\]

The attribute value that provides the smallest \( \text{Gini}_{\text{split}}(T) \) is chosen to split the node. For clearly explaining of random forest how to choose the best split is shown bellowed.

<table>
<thead>
<tr>
<th>Record</th>
<th>Attributes</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Subject</td>
<td>Verb</td>
</tr>
<tr>
<td>1</td>
<td>PN</td>
<td>SUCCEED</td>
</tr>
<tr>
<td>2</td>
<td>PS</td>
<td>ASSUME</td>
</tr>
<tr>
<td>3</td>
<td>PN</td>
<td>BRING</td>
</tr>
<tr>
<td>4</td>
<td>CN</td>
<td>NAME</td>
</tr>
<tr>
<td>5</td>
<td>CN</td>
<td>ASSUME</td>
</tr>
</tbody>
</table>

Table 1. Example dataset of Management Succession Domain.
This example shows the construction of a single tree using the abridged dataset. Only two attributes are chosen for this tree construction. CART splits the nodes by using a binary method where each decision class id to determine occurrence of NO or YES. There are five distinct records with values for two attributes, Subject and Verb.

Assume that the first attribute to be split is Subject attribute. The possible splits for Subject attribute in the left node range from \{PN,CN,PS\} ∈ x, where x is the split value. All the other value at each split from the right child node. The possible splits for the Subject attribute in the dataset are Subject ∈ PN, Subject ∈ CN, Subject ∈ PS. Taking the first split, the gini is calculated by using the above equations.

\[
\text{Gini}_{\text{Subject} \in \text{PN}} = 0.5 \\
\text{Gini}_{\text{Subject} \not\in \text{PN}} = 0.4452 \\
\text{Gini}_{\text{split}} = 0.46712
\]

The gini index for the Subject attribute at all possible splits are \text{Gini}_{\text{split(Subject} \in \text{PN)}} = 0.46712, \text{Gini}_{\text{split(Subject} \in \text{CN)}} = 0.46712 and \text{Gini}_{\text{split(Subject} \in \text{PS)}} = 0.4000. The lowest value of the gini index is chosen as the best split attribute. The procedure is repeated for the remaining attributes in the dataset. After that the single decision tree is built with the following rules:

- If Subject is PS, then class value is NO.
- If Subject is PN/CN and Verb is Assume, the class value is NO.
- If Subject is PN/CN and Verb is Succeed/Bring/Name, then class value is YES.

Random forest classifier follows this same methodology and constructs multiple trees for the forest using different sets of attributes. It has uses a part of the training set to calculate the model error rate by the out of bag (OOB) error estimate. The OOB error estimate affirms the model stability and supplies a weight for each tree. When the forest is employed classification, each individual tree votes for one class and the forest predicts the class that has the plurality of votes. For above example, we consider 6 random trees with the records if Subject is PN and Verb is Name, the class value is NO or YES. The weight voting is conducted by summation of the weight values for respective class and the class with the highest values is chosen as the best class. For the above rule, the class YES which has the most votes among all the cases is chosen over NO. Random forests have some interesting properties. They are more efficient since only a sample features need to be tested in each node, instead of all features; they are relatively robust to outlier and noise, and they are easily parallelized. The efficiency gain makes random forests especially interesting for handling a large number of features, may of which are expensive to compute.

4. Using Random Forest Induction as valid classification model

We show that random forest is a good choice for information extraction tasks as it runs fast on large and high dimensional databases. It is easy to tune and is highly accurate, outperforming popular algorithms such as decision trees. There are two methods for estimating classifier accuracy: the hold out method and k-fold cross validation methods. In this research, we used 10 folds cross validation method. The use of such techniques to estimate classifier accuracy increased the overall computation time, yet is useful for selecting among several classifier. For decision tree we used the C4.5 by Quinlan [57]. As decision tree exhibited poor classification accuracy of 49% in initial tests, we did not include them in further extraction experiment. Random Forests are classifiers that are constructed from a combination of decision trees. The single trees are trained on randomly drawn sample of the training data and the split variables during tree construction are chosen from a drawn subset of all variables. Averaging over these varying trees a better generalization is achieved compared to single decision trees. According to the accuracy rate of the classifier, we determine which classifier is used in our system. There are general techniques for improving classifier accuracy: bagging and boosting. We only look at only one of these two techniques. Given a set \( S \) of samples bagging works as follows. For iteration \( t (t = 1,2,..,n) \), a training set \( S_t \) is sampled with replacement from the original set of samples \( S \). Since sampling with replacement from the original set of samples of \( S \) may not be included in \( S_t \), while others may occur more than once. A classifier \( C_t \) is learned for each training set \( S_t \). To classify an unknown sample, \( x \), each classifier \( C_t \) returns its class prediction, which counts as one vote. The bagged classifier, \( C \), counts the votes and assigns the class with the most votes to \( x \). Bagging can be applied to the prediction of
continuous values by taking the average value of each classifier prediction. In this paper, we compared the accuracy rate of C4.5 and Random Forest on our data set.

We find that the accuracy rate of Random Forest algorithm is higher than that of the C4.5. So we decide to choose the Random Forest algorithm that used in our extraction engine. Random forest is one of the most successful tree based classifier. It has proven to be fast, robust to noise with our data set. Our approach extends the original idea of Random forest to deal with the data sparseness problem encountered in information extraction. Once constructed, the random forest’s function as a randomized history clustering which helps in dealing with the data sparseness problem. In general, the weakness of some trees can be compensated by other trees. After an extraction model is constructed, it can perform extraction on a given sentence by classifying candidate entities in the sentence into responsible category. In the extraction step, a candidate entity is classified as desired category when the random forest classifier returns a positive score value.

5. Developing Classification based Information Extraction System

In this paper, we describe a general method for building an information extraction engine using extraction features along with supervised learning algorithms. In this method, the extraction decisions are lead by the classifier instead of sophisticated linguistic analyzer. Whereas, linguistic analyzer restricts the kind of extraction patterns that can be learned by IE system, a broad range of problems can be formulated in terms of classifications. An instance might represent a situation and the classes might be several alternate actions that could be taken depending on the situation. In term of our information extraction engine’s concept definition, this means specifying a mapping between slots in the target concept and syntactic constituents of the instance. The proposed information extraction system uses a machine learning techniques to acquire the extraction model. The extraction model identifiers target entities by examining their features mix that includes those based on syntactic, semantics and others. The extraction process is very much a classification task that involves accepting and entity or combination of entities as target entities. Like other IE method, we divide our proposed Information Extraction System into two steps: the learning step and the extraction step. The former learns the extraction model for the target entities in the desired semantic category using the training sentences and their answer keys. The later applies the learnt extraction model on other sentences and extract new target entities.

The learning step consists of the following smaller steps.

(1) Sentence Parsing: The train sentences are processed by a sentence analyzer to produces instances for the information extraction engine. These instances have been segmented into syntactic constituents such as subject, verb, object and prepositional phrased. In addition, each word has been tagged with a semantic class. For corporate positions and corporate name, all positions and organization tagged that are used as candidates. For person in and person out, as there are usually more clues within a sentence indication a person as in or out. The candidate entities from the training sentences are defined as positive entities if their corresponding noun phrases match the syntactic constraints.

(2) Feature Acquisition: This step refers to deriving features for the training target entities that is the noun phrases.

(3) Extraction Model Construction: This step refers to constructing the extraction model using some machine learning techniques. The classification step performs extraction model using the learnt extraction model following the steps bellows.

(1) Sentence Parsing: The test sentences are parsed and defined as candidate entities in the responsible syntactic phrased to predict the target entities.

(2) Feature Acquisition: This step is similar to that in the learning step.

(3) Classification: This step applies Random Forest classifier to extract the candidate entities.

By identifying all the noun phrases and classifying them into positive entities,
according their categories, we transform the IE problem into classification problem. To keep our method simple, we do not use coreferencing to identify pronouns that refer to the positive entities. The following examples are prototype sentences and converting stages.

Our system segments each sentence into the parts of speech tags, parses the sentences, and derives features for the entities within the sentences. The features of the entities are derived from the syntactic structure of the sentence in which the candidate entities are found. And then we input the feature sets to the random forest classifier. The following procedures are converting the Sentences into the feature sets for the classifier and the result output of the system deal with the input sentences.

1. Alan G. Spoon succeeds Mr. Graham as president of the company.
2. David Culver retired chairman at Alcan Aluminium LTD.

We are carried out in the domain of Management Succession, representing free text, is a collection of Wall Street Journal articles. The target concept is a four-slot relation of corporation, position, person moving into that position and person moving out of the position. We used the same training and test data, and the some scoring criteria as pattern based extraction system developed by Soderland [14]. These dataset contain 6915 training instances and the test data are sentences extracted from 100 test documents, comprising 2840 instances within a total of 169 templates, 84 PersonIns, 100PersonOuts, 148 Positions and 92 Organizations.

<table>
<thead>
<tr>
<th>Sentence No</th>
<th>Person_In</th>
<th>Person_Out</th>
<th>Position</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alan G. Spoon</td>
<td>Mr. Graham</td>
<td>President</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>David Culver</td>
<td></td>
<td>chairman</td>
<td>Alcan Aluminium LTD</td>
</tr>
</tbody>
</table>

Table 2. Precision and recall for Random Forests Induction on the Management Succession Domain.

From the above experimental results, we found that the proposed extraction system reaching

---

Figure 2 Architecture of Proposed Information Extraction System
an F-measure as high as 87.5% by using the right choice classifier for the proposed system.

6. Conclusion and future work

In recent years, the emphasis on IE has shifted to adaptive IE. We feel that a classification approach allows systems to adapt to new domain by using a standard set of features. We have successfully transformed IE into a classification problem and adopted Random Forest Algorithm to extract target entities. We have not come across any paper reporting such an IE approach. We present a system in this paper that attempts IE on a sentence by sentence basis. In the future work, we will attempt to create the full scale scenario template task of information extraction system from free text.

References


Appendix:

SUBJ= Subject
VB= Verb
OBJ= Object
PP= preposition
PP-WORD=Preposition word
REL-O=Relative Object
REL-O-VERB=Relative Object of Verb
IE=Information Extraction System
PN=Person Name
PS=Position
CN=Company Name