The Development and the Evaluation of a System for Extracting Events from Web Pages

Mihai-Constantin AVORNICULUI, Silviu Claudiu POPA, Constantin AVORNICULUI
Babeş–Bolyai University Cluj-Napoca,
Faculty of Economic Science and Business Administration
{mihai.avornicului, silviu.popa, constantin.avornicului}@econ.ubbcluj.ro

The centralization of a particular event is primarily useful for running news services. These services should provide updated information, if possible even in real time, on a specific type of event. These events and their extraction involved the automatic analysis of linguistic structure documents to determine the possible sequences in which these events occur in documents. This analysis will provide structured and semi-structured documents in which the unit events can be extracted automatically. In order to measure the quality of a system, a methodology will be introduced, which describes the stages and how the decomposition of a system for extracting events in components, quality attributes and properties will be defined for these components, and finally will be introduced metrics for evaluation.

Keywords: *Event, Performance Metric, Event Extraction System*

1 Introduction

In the Internet world the quantity of information reached very high levels. To find specific information it was required the existence of tools (search engines) which automatically takes a scroll of existing pages to update their databases with the latest information on the Internet. Most of the times the search is based on a string in the web pages stored in the database search engine. The results of such searches are a large number of links to those pages.

To systematize the search and get a result in a tangible form, is useful another step processing the information returned by the search engine and generating responses in a more organized form.

Let's take the example of sports events. A user of this news service may want to know why sports games are conducted in a region of the world (Continent, Country, City etc.) Within a certain time: at that time, one day ago or next week, etc. All this information must be obtained from data already centralized. It can be obtained data about an event from multiple sources (web sites that pull information). Also source can complement each other in content information about certain events.

In the literature, the event is defined in different ways, depending on what is desired from an application to extract the events. Pustejovsky and his colleagues [2] define an event as something that happens from beginning to end for a particular document.

The event is a term (entity), which covers a situation that happens or appears to be punctual or refer to a specific period of time. Events are generally expressed by verbs, adjectives and predicative sentences.

In the example below the event is marked in bold.

In 221 BC, the first Emperor of China, Qin Shihuangdi, **conquered** the rest of China after a few hundred years of disunity.

Of course, the event is a series of attributes that will have to be identified.

A document from our point of view is a sequence of words. It is not a set of words because similar words like may occur in different places in the text and may belong to different categories of tags. For example the word "Paris" sequence can be an actor in "Paris Hilton" sequence, but also a location in the sequence "stayed at the Fashion Week in Paris. To resolve this ambiguity we use a set of learning concepts and attributes that will contain it.

Definition 1 We define a document $\mathscr{D}$ as a sequence of words c1, c_2 , …, c_m, where $m \in \mathbb{N}^*$.

Definition 2 A segment T = $t_1, t_2, \ldots, t_n$, it is a sequence of words in a document, which represents a sentence or a phrase and it can be seen as a atomic information entity where $n \in \mathbb{N}^*$.

Definition 3 We define a fragment F, as a sequence of segments, like $t_1, t_2, \ldots, t_n$, where $n \in \mathbb{N}^*$.

For identification we use sets of learning events. Next we'll define the set of learning.

Definition 4 A set of learning S is a series of examples of form: S = $s_1, s_2, \ldots, s_n$, where s_i = (*subject*, *action*, *value_list*), where *value_list* depends on searching domain. Each element from given example (*subject*, *action*, *value_list*) has the form <name_attribute> *value* </ name_attribute >.

Let's have the following example:

In 221 BC, the first Emperor of China, Qin Shihuangdi, conquered the rest of China after a few hundred years of disunity.

In this case the set of learning has the following form, specifying the name of the attributes:

SUBJECT: **history**
ACTION: Tag for attribute: **<action>** Extracting: **'conquered'**
LOCATION: Tagfor attribute: **<location>** Extracting: **'China'**
DATA: Tag for attribute: **<data>** Extracting: **'221 BC'**
ACTOR: Tag for attribute: **<actor>** Extracting: **'QinShi'**

Fig. 1. Example for learning set

A set of learning contains a number of such items for different areas. Based on this set of learning events can be extracted from original documents.

Definition 5 We consider F a lot of fragments generated a lot of documents, and A = {$A_1, A_2, \ldots, A_n$} a lot of attributes, where for A_i defines a set of values in the set of learning, $D_i = dom(A_i)$, i = 1, 2, …, *n*. The attribute A_1 contains each time the subject, and the attribute A_2 the action. We define $e_{Ssubiect}(f)$ the event obtained from the fragment F, using the learning set S, where $e_{Ssubiect}: F \rightarrow D_2$ and $e_{Ssubiect}(f) = v_2$, $v_2 \in A_2$, A_2 being the lot of values for action from the learning set.

Thus in the example above we can identify the event **conquered**.

Definition 6 We consider F a sum of fragments, and S = $s_1, s_2, \ldots, s_n$ learning set, s_i = (domain, action, value_list), $e_j = (subject, action, v_1, \ldots, v_{pj})$. Using S we generate D_{dom} using subject S and D_{act} on searching action. Considering $\mathcal{A}$ a sum of attributes, we define function h_e Using S we extract the events for a particular area and action, F fragment obtained from using the learning set, where $h_e: D_{dom} \times D_{act} \rightarrow \mathcal{P}(\mathcal{A})$ and h_e (*domain*, *action*) = e_j as each v_i to be included in F, where $\mathcal{P}$ is the sum of parts of A.

If we set the learning stated above and the following passage, we get an event structure, which will include event date, event and place of the actor.

If h_e (*domain*, *action*) has the structure means that we have identified the event.

Example for the above fragment:

h_e ('history', 'conquered') = ('history', 'conquered', 'Qin Shihuangdi', 'China', '221 BC').

This system is aimed at achieving a knowledge extraction system – events in HTML documents. The system will recognize the events of a certain type (weather, sports, politics, text data mining, etc.) depending on how they will be trained (dictionary of concepts that is). These events may be provided to the user or it can extract the entire context in which the event appeared to indicate that the initial event was incorporated [5].

This system aims to be a real help for the query information from several websites, when you want to identify events of a particular type. For example, if you want to find out more information about the weather websites (Yahoo, Google, etc.) For a given area and period, then the system would be helpful. For each type of event (weather, sports etc.) There shall be a dictionary of concepts. Construction of a dictionary concept is a task that requires more work. Trying to achieve a system that is extensible, so as to permit the extension or adding a new dictionary, the extraction of its specific events.

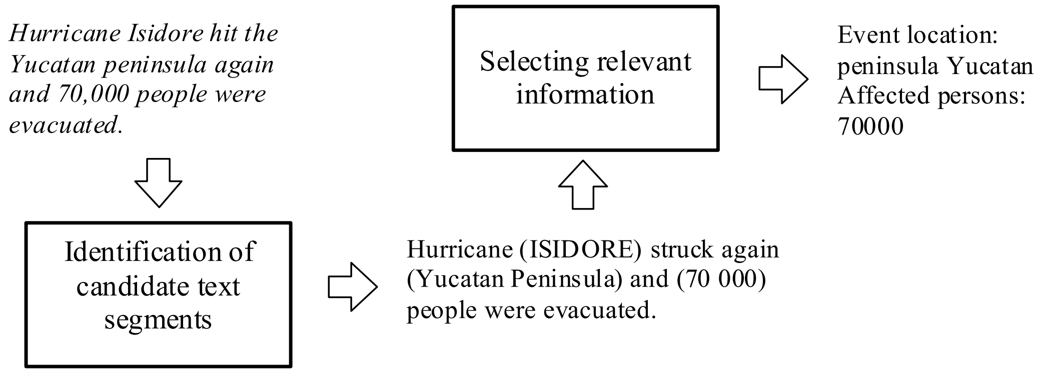
Hurricane Isidore hit the Yucatan peninsula again and 70,000 people were evacuated.
Identification of candidate text segments
Hurricane (ISIDORE) struck again (Yucatan Peninsula) and (70 000) people were evacuated.
Selecting relevant information
Event location: peninsula Yucatan
Affected persons: 70000

Fig. 2. The information extraction by text classification

If we want to extract relevant information from text documents, is enough to look at combinations of words around the desired information (e.g. background) to learn patterns for extracting necessary. Therefore we have two main stages:

A. Identify all segments, which may be part of the result.

B. Selection of candidate set those segments that are useful to us.

The following figure illustrates this process through a simple example presented in the news about a hurricane.

A. Identification of candidate segments

The purpose of this step is to detect most, if not all segments of text to be included in the result. Most information extraction systems take into account only simple evidence, while our system focuses on the detection of events with their place of deployment, the date and time of deployment, if necessary.

In order to identify candidate text segments, we can use a regular expression analysis. This type of analysis is general, robust and produces a high level of recall.

Figure 2 presents the first part of this phase. Words written in capital letters correspond to segments of text entry candidate.

B. Selecting relevant information

The aim of this phase is to capture segments of text, which must be part of the result. Classification is based on supervised learning techniques. In this context, each candidate text segment is classified according to its lexical context.

In contrast to the previous stage, the selection of relevant information must be more to achieve high accuracy than a recall. This motivates us to use a learning method in order to specify a different classifier for each type of event. The second part of Figure 2 illustrates this point. In the above example we are interested in the event, place and persons affected.

2 Conceptual Model

To achieve model system (components) should deal generally of three parts [5]:

- bringing websites on the Internet and save them in a database to be further processed: The system must receive a series of web addresses that need to travel.
- processing documents and obtain the necessary information: processing documents and extract the necessary information is based on a dictionary of concepts that describe the types of events.
- giving users a way to access the information collected: Finally have given users access to information extracted. For example, sporting events, users can submit a list of them, ordered by the date on which they have or have occurred.

The Figure 3 presents the relevant structure of the system.

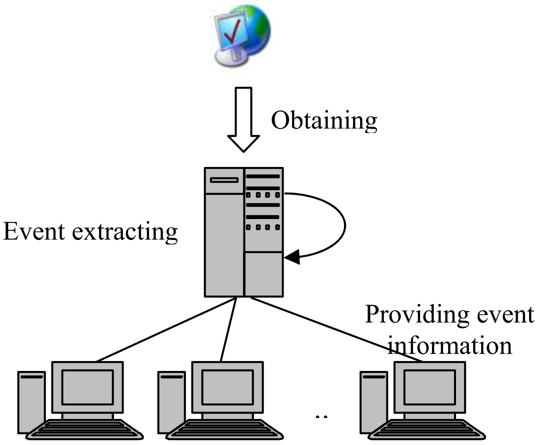

Fig. 3. The system structure

If users are human, will provide web pages with information you want, and if users are programs where information can be transmitted in a generic way, for example XML document.

3 Implementation

I used technology for implementation of Java Server Pages (JSP). JSP is the most popular way to create web interface for applications running on the Java platform, created by Sun. It is based on technology called Java Servlets is actually a complement to it as easy as the idea of creating dynamic Web pages.

The central point of technology is the so-called JSP pages are basically text files that combine HTML with Java code descriptions. JSP pages managed and accessible through an application server.

It receives HTTP requests coming from a Web browser. If an application relates to a JSP page, that page and local process server based on its content dynamically generate an HTML page that sends the browser response.

Processing server-side JSP pages requires actually creating Java Servlet class that follows the rules written in JSP page and includes Java code in it. Workflow system extracts the events in Figure 4.

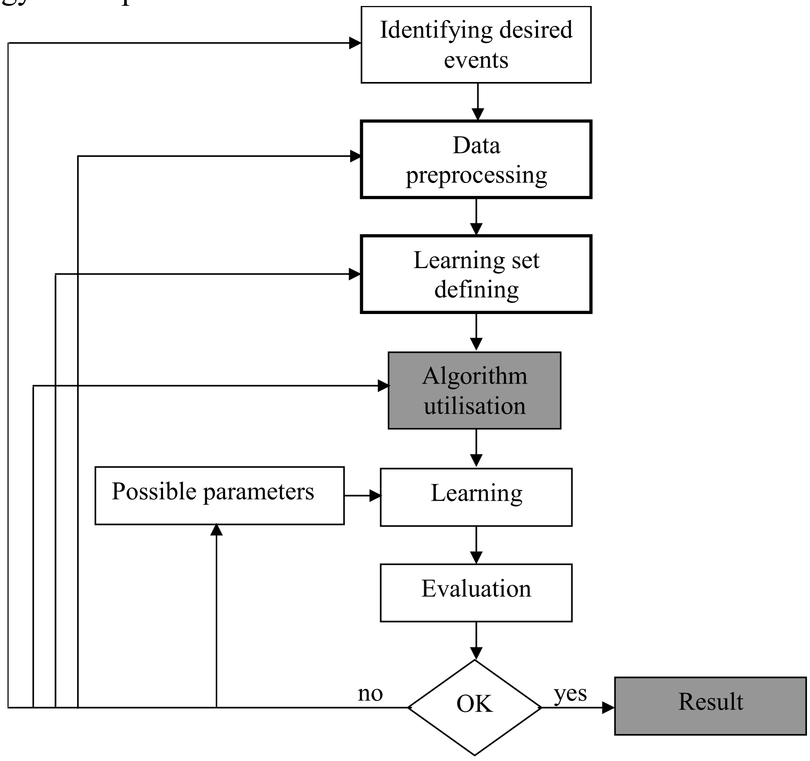

Fig. 4. Workflow within the system to extract events

In the literature, there are described many types of machines learning algorithms. We are interested in inductive learning algorithms that create situations of data. Inductive learning methods differ in knowledge representation (for example: rules, grammar) and search strategy, which are used to identify the assumptions of the data. Information retrieval systems based on learning algorithm can use two strategies: compression and covering. Systems that use compression, will perform a specific search in general and trying to compress sets of rules learned.

Event extraction system uses hedging strategy, and includes several strategies such as: AQ, CN2, spreadsheets, LLC.

The algorithm works as follows:

1. Starts with a vide set of rules.

2. Keep a positive instance, or a set of positive instances of learning set.

3. Find those rules, which can cover the rest of the court or courts.

4. Select the best rule of the proposed rules, on the basis of criteria which means the optimizing of generalization and compression.

5. Delete those instances that are covered by the training set.

6. If the training set is vide, we stop. If not, we repeat steps 2-5.

The algorithm is based on learning rules. Coverage algorithms can make the search more effective than compression, since it keeps the rules for those situations when it already covers a specific instance. Learned rules are more specific coverage algorithms, because there is a process to make the existing rules of general rules.

Learning algorithm is as follows:

1. Initialization rules with a default value (usually empty).

2. Initialization of all available examples.

3. Repeat

a. Finding the best rules on examples.

b. Set the example to all examples, if not handled correct set of rules.

Until you can usually find a better (e.g. until remaining examples).

For extracting the events, the first time we have to download the Web pages we want. The download addresses are specified for each area in an xml file. After this, it comes the learning of events.

These events and their extraction involved the automatic analysis of linguistic structure documents to determine the possible sequences in which these events occur in documents. This analysis will provide structured and semi-structured documents in which the unit events can be extracted automatically.

The function extractEventsForDomain:

```
Function
extractEventsForDomain(domainName)
Loading the database with files: files.xml
Loading the needed data for Name Entity Recognition
Loading the needed data for extracting events
Loading the database with events
If not exists this document
Then create a new one
End if
Search the node with current domain from eventDoc
Adding domain element with name attribute: <domain name=" ">
Adding the new node in xml
If founded the specified domain
Then
Keep the url of the current used file
Save the modified file by events
End if
End function
```

After generating semi-structured document, you can extract the properties of entities that describe an event. For example: event participants, location and time of an event. These entities were organized by classification and thus can extract semi-structured text data. Automatic classification of entities was performed using WordNet's.

The original idea was to use subclasses of class Event for events extraction from HTML documents. In this case it is easier to populate the ontology with court events, as it is not an enigma of whom belong the classes.

The ontology generation process for a particular type of event performs the following tasks:

- Learning sets are specified in an xml file (learning is a specific topic).
- Going to the point of interest, sub-class Event class in the new generation ontology are extended to include more keywords to events of a particular type.
- For each of these keywords, WordNet synonyms were used to strengthen a sub-class. In this way, repetition of concepts in the ontology can be avoided.

The strengthening the system is a hidden feature, so only the keyword of the sub-class

Event class is visible, although all events with keyword synonyms are stored in this subclass. We have such sub Slavery, which contains the following synonyms from WordNet as follows:

Slavery (*Romanian term*) = {Bondage, Slaveholding, Thraldom, Thrall, Thralldom}

Synonyms appear in brackets the word "slavery" in English. Used as synonym for class-class event is done manually. These concepts are then automatically generated by the JSP's WordNet.

The advantage of using WordNet to find synonyms' key event, such that differences WordNet cares and write in English the same word. For example: Civilization = {Civilization, Culture, Refinement}

The above example shows that Civilization can be written in the form of Civilization.

As mentioned above, the ontology system is composed of classes and subclasses Event events are specified keyword. Examples of class Event subclass could battle, disaster and the Treaty, to name a few.

Event ontology system properties include:

- exists – that play during the event, start and end date of the event
- hasCategory – indicating the category the event belongs to
- isRelationMemberOf – This category shows which other ontology belongs to a particular event.
- Related – this category lists all the items that depend on during the event, such persons and locations. It is the overall relationship, which is the parent of ontology other relationships. Indicate a connection between two examples.

The courts of other four subclasses of the ontology system were considered to discover the link between the event and other examples drawn from ontology. These four additional classes include action, actor, location and date. In general, for an event was limited to one sentence, because the search has assumed that if an event is mentioned in a sentence, then other relations can be identified in the same sentence. For example if "Overruned" (Employment) is found in a sentence, the sentence is most likely that gives information about the cause of employment, where employment was held, who attended when held.

4 Quality Assessment System

This section presents the approach to measure the quality of the extraction of events. To measure quality will be introduced a methodology that describes the stages and how the decomposition of a system for extracting the component events will be defined attributes and qualitative properties of these components (based on ISO 9126) and will ultimately be introduced metrics for their evaluation.

Since the approach used for implementation of events extraction systems is object-oriented further methodologies and principles used in quality evaluation of object-oriented systems will be analyzed. EMA (Capability Maturity Model) [4] is one of the top quality models used in software engineering, which proposes a unitary model for quality assessment.

Based on this model was established quality standard ISO/IEC 9126 that software quality factors fall into six categories: functionality, reliability, usability, efficiency, ease of maintenance and portability.

These six categories are divided into sub-measurable characteristics, which may provide clues about the overall quality system. Each sub-features quality is further divided into attribute quality, i.e. an entity that can be verified or measured in software. These attributes not defined in the standard because they can differ from one product to another or from one technology to another.

Recently, object-oriented paradigm has introduced other elements to estimate the quality of software that must be based on object-oriented principles: data encapsulation, inheritance and polymorphism. This approach led to the introduction of new quality metrics for estimating an object oriented system.

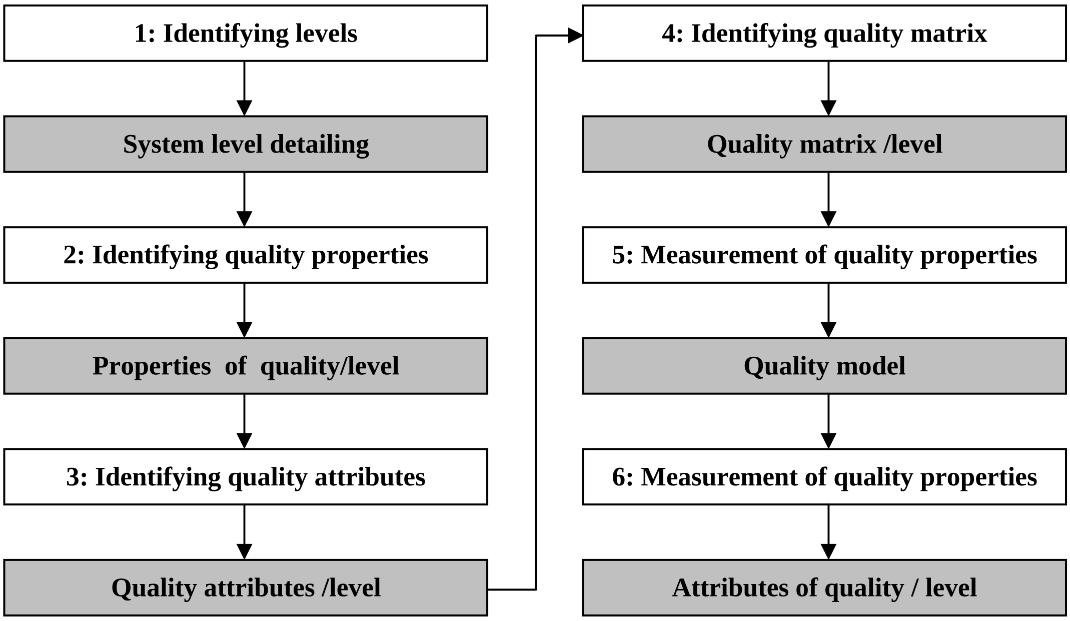
1: Identifying levels
System level detailing
2: Identifying quality properties
Properties of quality/level
3: Identifying quality attributes
Quality attributes /level
4: Identifying quality matrix
Quality matrix /level
5: Measurement of quality properties
Quality model
6: Measurement of quality properties
Attributes of quality / level

Fig. 5. Methodology for measuring quality

Based on ISO/IEC 9126, Bansya (2002) defined a new model consisting of hierarchical as many qualitative estimate attributes quality system design: re-use, flexibility, ease of understanding, functionality, extensibility and efficacy (effectiveness), and many of properties of system design: project size, hierarchies, abstraction, data encapsulation, coupling, cohesion, composition, inheritance, messaging (messaging) and complexity. These are combined using a qualitative matrix for linking adjacent levels, those with lower education. Regarding the quality of the analysis extracting events, progress in this area are relatively few. Most approaches focused on problem assessment, verification and validation systems. However, quality systems approach for extracting events from the perspective of ISO 9126 has not received much attention.

The start the purposes of quality assessment system for extracting events ISO was made by Nabil et al. (2005) study which will underpin the qualitative model proposed in this section. Since the structure of such a system is complex, the authors propose a methodology based on quality analysis of these systems break down and analyze attributes quality components for each component. In the second step, they are combined to build a model for overall quality system based on ISO 9126. The Figure 5 presents the components of this methodology, which is divided into six stages. The first four are qualitative phase to generate corresponding system matrices and the last two component standardization phase of this matrix.

Step 1 – Identifying levels. The first step is system analysis and identification of three levels [4]: database, model and model inference task. Peter a more accurate estimate of the quality, the database can be decomposed into tables.

Step 2 – Identify qualitative properties. Qualitative properties proposed in ISO 9126 and quality attributes proposed for object-oriented systems were analyzed and reviewed to determine whether and to what extent contribute to the qualitative aspects of the event extraction system. For each component identified in the first stage have been defined a lot of quality properties:

- Database – coupling, redundancy, updateability, consistency, robustness, complexity, cohesion
- The inference – redundancy, robustness, cohesion, complexity, modifying, composition, coupling, completeness

- The task – redundancy, robustness, cohesion, complexity, modifying, composition, coupling

Step 3 – Identifying quality attributes. In the third stage, attributes define quality. Proposed attributes Bansya (2002) for object-oriented systems were preferable to those proposed in ISO 9126 because characterized best extraction system events. Thus quality attributes for each component were distributed as follows:

- Database – functionality, ease of comprehension, reliability, extensibility, efficiency
- The inference – reuse, flexibility, ease of understanding, functionality, extensibility and efficiency
- The task – reuse, flexibility, ease of understanding, functionality, extensibility and efficiency

Step 4 – Identification of qualitative matrices. Using qualitative attributes and properties defined in previous phases, qualitative matrices are defined for each system component. To mathematically how this affects properties (positive or negative), the value of each quality attribute is defined as the weighted average property values:

$$A_i = \sum_i w_i p_i \qquad (1)$$

where A$_i$ represents an quality attribute (reuse, flexibility, etc.), p$_i$ represents a property (completeness, cohesion, complexity, etc.) and w$_i$ share that property. The weights satisfy the following restrictions: $\forall w_i : -0.5 \le w_i \le 1 \;\; \sum_i w_i = 1$.

Step 5 – Establish quality model. At this stage the system is determined for each property values and based on these values and calculated values of qualitative matrices quality attributes. The model thus obtained is stored in a base model. The quality models developed in the previous stage for such a system components are stored in a base model for all systems evaluated.

Step 6 – Determine the final determination on quality. For each system generates a final assessment on the quality by comparing the qualitative model of the system, not the values determined for previous systems.

4.1 Experimental Rating System

We use 350 types of learning experiments after eliminating types that do not contain relevant information.

Assessing the system is to determine the relationship between different parameters and system performance, while the comparison is to compare WHISK system and other systems. Similar information recovery, precision and recall are used as your performance metric extraction systems of events (or information). Precision (precision) is the percentage of information extracted from the system properly, while your recall (recall) is the percentage of relevant information that can be extracted correctly by the system.

For facilitating the comparison of system performance, we use a F measure [3], which combines precision and recall into a single measure of extraction of information (the event).The *F* measure is defined in the following equation :

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \qquad (2)$$

where, *P* is precision, *R* is recall and *β* is a parameter that measures the relative importance of precision and recall. *B* is usually 1. Measure *F* in event extraction systems is approximately 0.6 [3] [7].

To some extent, the number of extraction rules induced by a set of learning may reflect the complexity generated extraction patterns and structure inherent in the learning set. Compact extraction rules can help users understand the basic regularity of the field.

To test the number of courts to initiate the event extraction system performance, we can obtain an item from one court to initiate a random sampling. Measure an item begins at 25 goes to court and all courts in the learning set. For each measure of learning, we report mean and standard deviation of five samples in a separate set of tests with 350 instances. Performance metrics reported include your precision, recall site, and the number of rules. Table 1 presents the system performance when the number of instances of learning gains.

Table 1. The System's performance in sports

Opening measure	Precision+/- StdDev(%)	Recall+/- StdDev(%)	#Rules +/- StdDev
25	82.85+/-1.56	79.42+/-2.08	7.6+/-1.151
50	87.17+/-1.26	85.11+/-1.13	13.2+/-1.084
75	89.72+/-1.31	84.39+/-1.25	13.4+/-0.975
100	88.81+/-0.79	85.46+/-1.14	20.8+/-1.342
125	91.69+/-0.53	88.39+/-1.34	20.6+/-1.823
150	92.62+/-0.54	90.70+/-0.71	24.0+/-2.915
175	92.60+/-0.73	90.36+/-1.07	28.6+/-1.823
200	92.34+/-0.52	90.99+/-1.43	31.0+/-1.768
225	92.66+/-1.02	91.07+/-1.22	34.0+/-1.658
250	93.38+/-0.24	91.63+/-0.97	35.4+/-1.997
275	93.46+/-0.24	92.82+/-0.16	37.8+/-2.074
300	93.31+/-0.18	92.93+/-0.21	39.8+/-0.652
325	93.73+/-0.08	93.20+/-0.08	43.4+/-0.758
350	93.85+/-0.00	93.31+/-0.00	44.4+/-0.447

An interesting thing is that what really matters for an extraction system is the number of instances of learning events that have different syntactic structures surrounding the relevant passages of text. This could define non-monotony in table 5.5, increases as the measure of initiation.

Precision's system increased from 0.8285 to 0.938 and from 0.7942's recall in 0933 when the extent of open courts increased from 25 to 350 instances, compared with WHISK, whose precision has increased from 0.85 to 0.92 and from 0.83's recall to 0.94 when the extent of initiation increased from 25 to 400 instances [7], and compared with the system come out, whose precision has increased from 0.87 to 0.93 and Recall from your 0.85 to 0942, when the extent of initiation increased from 25 to 500 instances [3].

Taking into account that our system used random samples while the samples used by iASA and WHISK used selective, the performance of events extraction system is in competition with that of WHISK's and iASA.

5 Conclusions

The system proposed in this article was implemented as a functional prototype in Java using the Eclipse development environment. Development of this prototype raises interesting issues such as analysis, design, and as new extensions of event extraction systems. The system is functional and medium term is to extend the system with new features.

We proposed an algorithm for extracting events and experiments have proved effective algorithm. It was proposed to initiate a new strategy rules that give priority to the most specific rules. The system has been verified and validated.

Acknowledgement

This work was supported by Romanian National Authority for Scientific Research under the grant no. 91-049/2007 (PN2)

References

[1] B. Bai, J. Weston, D. Grangier, R. Collobert, Y. Qi, K. Sadamasa, O. Chapelle, and K. Weinberger. *Learning to rank with (a lot of) word features*. Information Retrieval – Special Issue on Learning to Rank, 2009.

[2] J. Pustejovsky, B. Ingria, R. Sauri, J. Castano, J. Littman, R. Gaizauskas, A. Setzer, G. Katz, and I. Mani. *The Specification Language TimeML*, The Language of Time: A Reader. Oxford University Press, 2005

[3] J. Tang, J. Li, H. Lu, B. Liang, and K. Wang. *iASA: Learning to annotate the*

semantic Web. Journal on Data Semantic, 2005

[4] K. Balla, *Minőségmenedzsment a szoftverfejlesztésben*, Panem, Budapest, 2007

[5] M. Avornicului, *Aspect of using UML for Designing an Event Extraction System*, publicată în volumul Education, Research & Business Technologies, Bucureşti, România, 2009

[6] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. *Open information extraction from the Web*. In Proceedings of the International Joint Conference on Artificial Intelligence, 2007.

[7] S. Soderland. *Learning Information Extraction Rules for Semi-Structure and Free Text*, in Machines Learning Journal, 1999

[8] S.B. Kotsiantis. *Supervised Machine Learning: A Review of Classification Techniques*, Informatic, an International Journal of Computing and Informatics, 2007

[9] T. Y. Lin, Y. Xie, A. Wasilewska, C.J. Lian, *Data mining: Foundations and Practice*, Springer Berlin, 2008

[10] S. S. Anand şi B. Mobasher. *Contextual Recommendation*, in *From Web to Social Web: Discovering and Deploying User and Content Profiles*, Lecture Notes in Computer Science (LNCS 4737), pag. 142-160, Springer Berlin-Heidelberg, 2007

Mihai-Constantin AVORNICULUI has graduated Faculty of Mathematics and Computer Science, Babes-Bolyai University Cluj-Napoca in 2004. He has finished his PhD studies in 2009. He works at the Business Information Systems department of FSEGA, Babes-Bolyai University Cluj-Napoca. His research interests include data mining, information systems, and advantage database systems.

Silviu Caludiu POPA has graduated Faculty of Economics, Babes-Bolyai University Cluj-Napoca in September 1997. He has finished his PhD studies in 2003.He works at the Business Information Systems department of FSEGA, Babes-Bolyai University Cluj-Napoca. His current position is PhD Lecturer.

Constantin AVORNICULUI has graduated Faculty of Economics, Babes-Bolyai University Cluj-Napoca in 1972. He has finished his PhD studies in 1992. He works at the Business Information Systems department of FSEGA, Babes-Bolyai University Cluj-Napoca. His research interests include information systems, and advantage database systems.