Using a Cellular Automaton to Extract Medical Information from Clinical Reports

Fatiha Barigou*, Baghdad Atmani* and Bouziane Beldjilali*

Abstract—An important amount of clinical data concerning the medical history of a patient is in the form of clinical reports that are written by doctors. They describe patients, their pathologies, their personal and medical histories, findings made during interviews or during procedures, and so forth. They represent a source of precious information that can be used in several applications such as research information to diagnose new patients, epidemiological studies, decision support, statistical analysis, and data mining. But this information is difficult to access, as it is often in unstructured text form. To make access to patient data easy, our research aims to develop a system for extracting information from unstructured text. In a previous work, a rule-based approach is applied to a clinical reports corpus of infectious diseases to extract structured data in the form of named entities and properties. In this paper, we propose the use of a Boolean inference engine, which is based on a cellular automaton, to do extraction. Our motivation to adopt this Boolean modeling approach is twofold: first optimize storage, and second reduce the response time of the entities extraction.

Keywords—Clinical Reports, Information Extraction, Cellular Automaton, Boolean Inference Engine

1. INTRODUCTION

An important amount of clinical data concerning the medical history of a patient is in the form of clinical reports that are written by doctors. They describe patients, their pathologies, their personal and medical histories, findings made during interviews or during procedures, and so forth. CR presents a source of precious knowledge that can be used in several applications, such as research information to diagnose new patients, epidemiological studies, decision support, statistical analysis, and data mining, but these CR are difficult to analyze due to their unstructured nature and the large volume of records available. Thus, an efficient system for extracting information in structured form can greatly benefit these applications. These kinds of systems have become very necessary tools; they will enable researchers to access the accurate data and required information, and reducing the time spent by doctors on making decisions about the patient. At our laboratory, we are interested in developing a system of mining textual data from clinical reports. The system is typically comprised of two main phases. We first, use the techniques of natural language processing to extract the relevant information to be stored in a structured database from clinical reports. In the second phase, the data mining tools will be used to

Manuscript received May 2, 2011; first revision December 14, 2011; accepted January 25, 2012.
Corresponding Author: Fatiha Barigou
* Dept. of Computer Science, Faculty of Sciences, University of Oran, Algeria ({fatbarigou, atmani.baghdad}@gmail.com, bouzianebeldjilali@yahoo.fr)

extract information that will be used in the exploration and discovery of new knowledge. In this paper, we focus on the first phase; and we are interested in extracting entities from French clinical reports (FCR). Most of the elements in these reports are name entities (e.g.., the names of patients, diseases, symptoms, and drugs).

There are three major reasons why we felt it was important to capture entities from clinical reports.

First, we know that these reports represent a source of valuable knowledge that can be used in several applications, such as research information to diagnose new patients, epidemiological studies, decision support, statistical analysis, and data mining. But these types of CR are difficult to analyze due to their large volume and unstructured nature.

Second, we believe that such a system for extracting entities can greatly benefit these applications. It will enable researchers to access the required information, and will reduce the time spent by doctors on making decisions about the patient.

Third, most of the work on named entity extraction (NEE) has been done in English. Research being done in French medical language is only in the initial stages.

For these reasons, we propose an original approach to do extraction from French clinical reports. We propose the CASI cellular automaton as a tool for extracting entities. We will study how to adapt this automaton to extract the named entities from French clinical reports.

The rest of this paper is organized as follows: Section 2 summarizes the named entity extraction task and work related to this field. Section 3 presents the cellular automaton CASI, which is a cellular inference engine and a machine learning system. In section 4 we propose to study our approach based on this cellular automaton, which is a contribution to the improvement of rule-based methods in the field of medical information extraction. Section 5 presents evaluation results concerning our proposed system performance. In Section 6 we conclude this research paper with the motivations that led us to adopt the principle of Boolean representation and Boolean inference to extract named entities and finally, we present some ideas for future works.

2. RELATED WORK

Named Entity Extraction consists in the automatic determination of continuous fragments of texts (called Named Entities), which refers to information units such as persons, geographical locations, names of organizations, dates, percentages, amounts of money, and references to documents [21].

Named entities extraction is mostly based on two different groups of methodologies: (a) pattern matching rules and (b) machine learning (ML). Many systems combine both approaches but the majority does not use machine learning and instead rely only on pattern matching, rules, and dictionaries [20].

The first approach is to manually define linguistic rules to detect each type of entity. These rules use lexical markers, dictionaries of proper names, and dictionaries of general language for identifying and typing named entities. This approach is time consuming during development, but gives very good results. The main disadvantage of these approaches is their lack of generalization, which limits their extension to new domains. Machine learning techniques have demonstrated remarkable results in the general domain and hold promise for medical information extraction, but they require large, annotated corpora for training, which are both expensive and

time-consuming in regards to training the models [20].

NEE has attracted the interest of many researchers, and there is an enormous amount of published research on this technology [7]. Although mostly tested on general entities such as names, places, organizations, dates, times, and numeric expressions [17], named entities extraction was also used, with promising results, in medical and biomedical texts to extract entities such as the names of genes, proteins, diseases, and symptoms. NEE has been applied to medical records and other clinical documents, such as reports from radiology and mammography. In [19] Meystre et al. presented a review of recent research on information extraction from medical records.

For rule-based approaches, there has been a large effort in processing clinical reports. Many clinical NLP systems have been developed, including MedLEE [12], SymTex [13], and MetaMap [1].

The MedLEE system [12] was applied to patient records using natural language processing techniques. It can extract useful entities from radiology and mammography reports to identify patients with tuberculosis [17] or breast cancer [14]. A similar approach was used in [6] for the automatic detection of fevers from clinical reports and therefore the possibility of detecting the existence of infectious diseases in affected patients… MedLEE [11] was even combined with machine translation to detect abnormal findings and devices in Portuguese radiology reports [5].

The authors in [23] presented an approach based on MetaMap for the extraction of medical entities of 20 medical classes from pathologist reports. The authors in [19] obtained 89.9% recall and 75.5% precision for the extraction of medical problems with an approach based on MetaMap Transfer (MMTx) and the NegEx negation detection algorithm.

Embarek and Ferret [10] proposed an approach relying on linguistic patterns and canonical entities for the extraction of medical entities belonging to five categories: disease, treatment, drugs, tests, and symptoms.

In the literature we found some studies that have worked on extracting only drug names from clinical reports. The representative research efforts include [8, 18, 27].

Chhieng et al. [8] reported a precision of 83% when using a string matching method to identify drug names in clinical records. Levin et al. [18] developed an effective rule-based system to extract drug names from anesthesia records and mapped them to RxNorm concepts with 92.2% sensitivity and 95.7% specificity. Recently, Xu et al. [27] developed a rule-based system for extracting medication information, called MedEx, and reported F-scores over 90% on extracting drug names, dosages, routes and, frequency in drug use from discharge summaries.

Recent systems are almost always based on some machine learning methods. An example is the semantic category classifier developed by [24]. It employs support vector machines to attribute semantic categories to each word in discharge summaries. In [9] they developed a medical information extraction system that combines a rule-based extraction engine with machine learning algorithms to identify and categorize references in clinical reports to patients who smoke.

The largest efforts to develop and evaluate information extraction from clinical text have been achieved in the context of the i2b2 smoking status identification challenge in 2006 and the Medical NLP challenge in 2007. A corpus of 502 de-identified and “re-identified” discharge summaries was first created by Uzuner et al. [25]. The task was to use discharge summaries to classify each patient as current smoker, past smoker, non smoker, or unknown. The best performing system was developed by Clark et al. [9].

The authors in [16] implemented a machine-learning-based named entity recognition system for clinical text and systematically evaluated the contributions of different types of features and

ML algorithms using a training corpus of 349 annotated notes. This project was part of the 2010 Center of Informatics for Integrating Biology and the Bedside/Veterans Affairs (VA) natural-language-processing challenge. Based on the results from training data, the authors developed a novel hybrid clinical entity extraction system, which integrated heuristic rule-based modules with the ML-base named entity recognition module. On a test corpus containing 477 hospital discharge summaries, they achieved an F-measure of 0.8391 for concept extraction and 0.9313 for assertion classification.

3. CELLULAR AUTOMATON CASI

CASI (Cellular Automata for Symbolic Induction) is a cellular method of generation representation and a means to optimize induction graphs generated from a set of learning examples [2]. This Cellular system is organized into cells where each cell is connected only with its neighbors (subset of cells). All cells obey in parallel to the same rule, which is called the "local transition function" This results in an overall transformation of the system.

As illustrated in Fig.1, CASI is composed of three modules: COG (Cellular Optimization and Generation), CIE (Cellular Inference Engine), and CV (Cellular validation) [2].

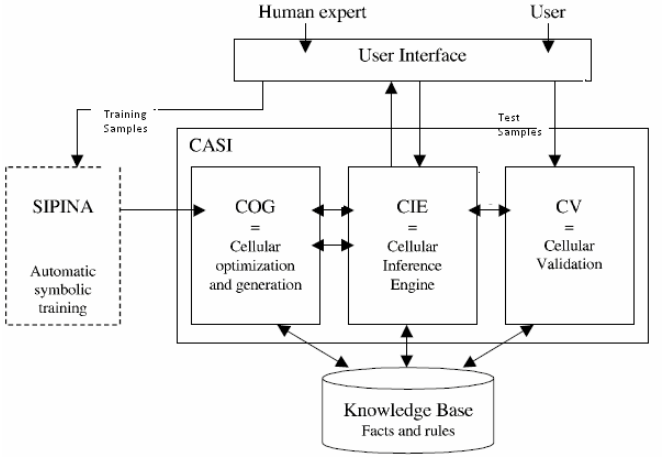

Fig. 1. Diagram Bloc of CASI [2]

3.1 COG module

Using a cellular automaton and cooperating with an induction graph (SIPINA method), COG module will extract new knowledge from training data. Two finite layers of finite automata represent the knowledge that is generated.

3.2 CV module

After the rules have been generated by the SIPINA method, which has been coupled along with the CASI machine, validation of this knowledge could be done using the CV module.

3.3 CIE module

In this work, we are interested by the CIE component. The authors in [2] consider CIE as a cellular automaton that is made of two finite arbitrary long layers of finite state machines (cells) that are all identical. The operation of the system is synchronous, and the state of each cell at time t+1 depends only on the state of its vicinity cells, and on its own state at time t.

This module, which is the core of the machine CASI, simulates the functioning of the basic cycle of an inference engine by using two finite layers of finite automata. The first layer, called *CELFACT*, is for representing the fact base, and the second layer, called *CELRULE*, is for representing the rule base. In each layer, the content of a cell determines whether and how it participates in each inference step. At every step, a cell can be active or passive, and can take part in the inference or not. The states of cells are composed of two parts: *EF* and *SF*, and *ER* and *SR*, which are the input and output parts of the *CELFACT* cells, and of the *CELRULE* cells, respectively.

Any cell *i* in the *CELFACT* layer with input *EF(i) = 1* is regarded as representing an established fact. If *EF(i) = 0*, the represented fact has to be established. Any cell *j* of the *CELRULE* layer with input *ER(j) = 0* is regarded as a candidate rule. When *ER(j) = 1*, the rule should not take part in the inference.

Two incidence matrices called RE and RS define the neighborhood of cells. They represent the facts input relation respectively and the facts output relation. They are used in forward chaining.

The input relation, noted iR_Ej, is formulated as follows: *if (fact i* $\in$ *Premise of rule j) then* $iR_Ej = 1$ *else* $iR_Ej = 0$.

The output relation, noted iR_Sj, is formulated as follows: *if (fact i* $\in$ *Conclusion of rule j) then* $iR_Sj = 1$ *else* $iR_Sj = 0$.

In order to illustrate the cellular inference engine, let us consider the set of rules generated by ML SIPINA (see Table 1). Table 2 shows how the automaton layers CELFACT and CELRULE represent the knowledge base.

As illustrated in Table 3, the neighborhood in CIE is defined by the incidence matrices of input (R_E) and output (R_S).

Finally, since there are *l* cells in the layer *CELFACT*, the *EF* and *SF* will be considered as *l-dimensional* vectors ($EF, SF \in \{0, 1\}^l$). Similarly, since there are *r* cells in the layer *CELRULE*, the *ER* and *SR* will be considered as *r-dimensional* vectors ($ER, SR \in \{0, 1\}^r$). Fig. 2 shows the general outline of the cellular automaton.

Table 1. An example of a knowledge base

R1	if (A and B) then C
R2	if (F and D) then A
R3	if (D and E) then B
R4	if (B and D) then F
R5	if (E and F) then D
R6	if (E and F) then B
R7	if (B and F) then G

Table 2. Cellular representation of a knowledge base. We have seven facts and seven rules

CELFACT	*EF*	*SF*
A	0	0
B	0	0
C	0	0
D	0	0
E	0	0
F	0	0
G	0	0

CELRULE	*ER*	*SR*
R1	0	1
R2	0	1
R3	0	1
R4	0	1
R5	0	1
R6	0	1
R7	0	1

Table 3. Input and output incidence matrices

RE	*R1*	*R2*	*R3*	*R4*	*R5*	*R6*	*R7*
A	1						
B	1			1			1
C							
D		1	1	1			
E			1		1	1	
F		1			1	1	1
G							

RE	*R1*	*R2*	*R3*	*R4*	*R5*	*R6*	*R7*
A		1					
B			1			1	
C	1						
D					1		
E							
F				1			
G							1

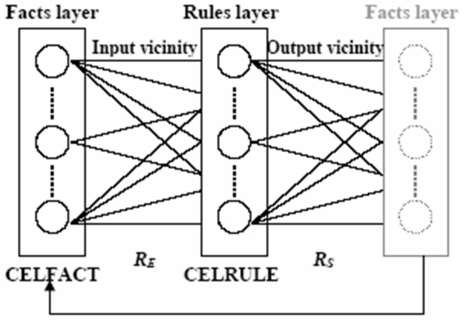

Fig. 2. Cellular Automaton for systems inference [2]

A goal fact, which is the basic cycle of an inference engine in forward chaining, traditionally operates as follows:

1. Search for applicable rules (evaluation and selection).
2. Choose one of these rules for example R (filtering).
3. Apply and add the conclusion part of R to the fact base (execution).

The cycle is repeated until the goal fact is added to the fact base, or stops when no rule is applicable.

The cellular automaton dynamics implements the CIE component as a cycle of an inference engine made up of two local transitions functions δfact (equation 1) and δrule (equation 2),

where δfact corresponds to the evaluation, selection, and filtering phases and δrule corresponds to the execution phase.

$$(EF, SF, ER, SR) \xrightarrow{\delta fact} (EF, EF, ER + (R_E^T \times EF), SR) \quad (1)$$

After applying this function we have:

$$EF = EF;\ SF = EF;\ ER = ER + (R_E^T \times EF)\ and\ SR = SR$$

$$(EF, SF, ER, SR) \xrightarrow{\delta rule} (EF + (R_S \times ER), SF, ER, \overline{ER}) \quad (2)$$

After applying this function, we have:

$$EF = EF + (R_S \times ER);\ SF = SF;\ ER = ER\ and\ SR = \overline{ER}$$

Where R_E^T is the transposed matrix of R_E and $\overline{ER}$ is the negation of ER

We consider G0 as the initial cellular automaton configuration (see Table 4) and the $\Delta = \delta rule \circ \delta fact$, as a global transition function: Δ(G0) = G1 where δfact(G0) = (G'0) and δrule(G'0)= G1

Let G = {G0, G1, . . .,Gq} be the configuration set of the cellular automaton. The automaton evolution in discrete time steps from one generation to the next and is defined by the configuration sequence G0, G1…Gq, where Gi+1 =Δ(Gi).

As an example, let us try first to establish fact C with the knowledge base from Table 1 where

Table 4. Initial cellular automaton configuration: G0. The EF of D and E are set to 1

CELFACT	EF	SF	CELRULE	ER	SR
A	0	0	R1	0	1
B	0	0	R2	0	1
C	0	0	R3	0	1
D	1	0	R4	0	1
E	1	0	R5	0	1
F	0	0	R6	0	1
G	0	0	R7	0	1

Table 5. Configuration obtained with δfact

CELFACT	EF	SF	CELRULE	ER	SR
A	0	0	R1	0	1
B	0	0	R2	0	1
C	0	0	R3	1	1
D	1	1	R4	0	1
E	1	1	R5	0	1
F	0	0	R6	0	1
G	0	0	R7	0	1

Table 6. Configuration G1 = Δ(G0) obtained with δfact • δrule(G0)

CELFACT	EF	SF	CELRULE	ER	SR
A	0	0	R1	0	1
B	1	0	R2	0	1
C	0	0	R3	1	0
D	1	1	R4	0	1
E	1	1	R5	0	1
F	0	0	R6	0	1
G	0	0	R7	0	1

Table 7. Final configuration G= {G_0, G_1, G_2, G_3, G_4}. Fact C established that its EF is set to 1

CELFACT	EF	SF	CELRULE	ER	SR
A	1	0	R1	1	0
B	1	0	R2	1	0
C	1	0	R3	1	0
D	1	0	R4	1	0
E	1	0	R5	0	1
F	1	0	R6	0	1
G	0	0	R7	0	1

D and E are initial facts (D, E ∈ fact base).

Initially, all the cell inputs in the CELFACT layer are passive (EF = 0), except those representing the initial facts (EF(1) = 1) (see Table 4). Using the cellular automaton principle, Table 5 presents the two layers, CELFACT and CELRULE, after evaluating, selecting, and filtering them in the synchronous mode with the first transition law, δfact. After the application of the second transition law, δrule, we obtain the configuration G1, as shown in Table 6.

δfact and *δrule* will be executed in parallel until goal *C* is reached or no rule is applicable. At the end we have the final configuration, as shown in Table 7.

4. CELLULAR APPROACH FOR EXTRACTING NAMED ENTITIES

We discuss in this section the use of cellular automaton for extracting named entities from clinical reports. Our goal is shown in Fig. 3[1].

The system proposed for this extraction task consists of two modules. The first is responsible for building the Boolean knowledge base following the principle of the cellular automaton CASI. The second module uses the inference engine CIE of CASI to classify named entities. In this paper we use the manually written rules that we have already established in [3, 4].

To extract named entities from French clinical reports written in a free and natural language, our contribution adopts the following approach:

- Manual construction of named entities classification rules;
- Boolean Modeling of constructed rules;

[1] Patient names in the report are not real names

il s'agit de la patiente Aîssati Lyla, âgée de 35 ans originaire et demeurant à El Bayadh, sans antécédent, mariée, mère de 2 enfants admise à notre service pour Brucellose le 25/07/2009.
Le début remonte 1mois par syndrome suderoalgique, sueur profuse nocturne, fièvre élevée avec arthralgie diffuse et asthme ayant consulté à plusieurs reprise d'où sérologie Wright demandé, d'où son orientation à notre service.
A l'examen clinique, patiente consciente, état général conserve Apyrétique,
TA normal, conjonctif normo colorée. Examen CV RAS, examen pleur pulmonaire
RAS, l'examen Abdominal normal.
Bilan fait retrouve FNS : GB 3700 , Ly 1.7 x 103/mm3, PN1.9 x 103/mm3,
Plaquette 100 x103/mm3, Urée , créatine 9.4 g , TGO 29, TGP 31.75 u/l.
Mise sous Gentamicine 16g pendant 10j, Doxycycline 100g 2cp/j
L'évolution était favorable.

Template Id File Reference:	ID-005 c\reports\report5.txt
Patient	Surname: Aissati Name: Lyla Age : 35 ans Adress: El Bayadh
Disease	Name : Brucellose Code CIM10: A23 Date of consultation : 25/09/2009
Symptom	Syndrome suderoalgique, sueur profuse nocturne, fièvre élevée …
Clinical Review	GB 3700, Ly 1.7 x 103/mm3, PN1.9x 103/mm3, ..
Medication	Name: Gentamicine 16g Duration: 10j Name: Doxycycline 100g Dosage: 2cp/j

Fig. 3. From unstructured text to structured information. The figure shows different types of entities (person, disease, symptom etc.) with their properties extracted from French clinical reports

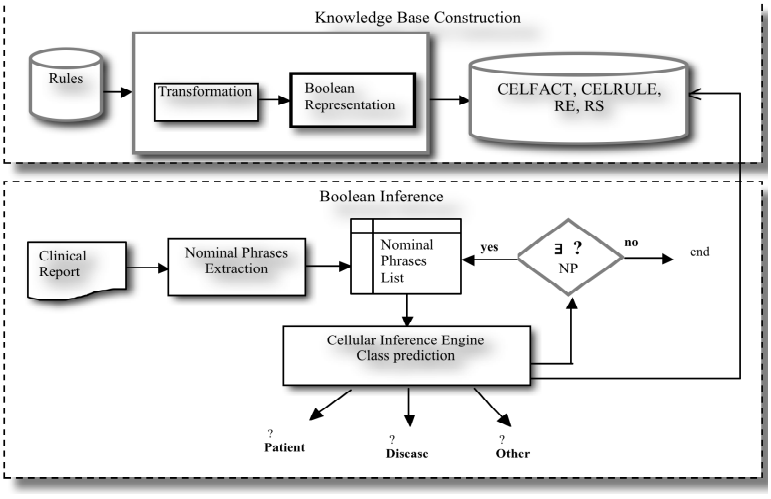

Fig. 4. System Architecture based on cellular automaton

- Linguistic Analysis of clinical reports for extracting nominal phrases with their morpho-syntactic and semantic properties; and
- Boolean Inference for classifying nominal phrases into different classes (person, date, symptoms, disease...).

4.1 Building rules for classifying entities

A preliminary study [4] on a corpus of medical records of patients in the infectious diseases department of the hospital of Oran allowed us to identify the different entities that must be extracted for future data mining. We selected the following entities: person, disease, symptom, medication, place, and date (consultation, birth, duration of treatment), and to study the syntactic structure of each type of named entity. Indeed, we found that most of the named entities present

Table 8. Nominal Phrases extracted from clinical reports and their classes

Term	*Grammatical sequence*	*Class*
Patiente Aissati lyla	NOM + NAM +NAM	Person
35 ans	NUM +NOM	Age
El Bayadh	NAM	City
Syndrôme suderoalgique	NOM + ADJ	Symptom
Gentamicine 16g	NAM + NUM + ABR	Medication

in a clinical report are terms (or nominal phrases: NP) composed of several words (see Table 8).

To recognize these nominal phrases (NP), we rely on their linguistic characteristics that are part of the grammatical categories of the NP and other syntactic rules of their arrangement. In Table 8 we present the grammatical sequences of some of the NP that we encounter most often in a FCR. We first tokenize and tag words that appear in different sections of (FCR) with the TreeTagger[2] tool. Secondly, the tagged medical report is parsed with a NP parser in order to detect NP. The analysis began by applying a set of syntactic rules to locate all the nominal phrases that are present in different sections of the clinical report. A filter is then applied to favor the longest NP among several NPs who share the same name.

To classify these NP, we decided to consider their internal structure and their neighbors that are on the left of NP. Thus, for our work, the process of named entities classification is to first recognize the NP that are present in the text of the clinical report, and then assign a class to these NP. To do so, we used the following four properties:

- Morphosyntactic information of the first three words of the NP.
- Semantic information of the first three words of the NP.
- Size of the NP.
- Neighborhood of the NP.

Thus, each NP is represented by a vector of words; these words are those belonging to this NP and those that appear in its neighborhood on the left. For example, the term "patiente Aissati Lyla"[3] of the report shown in Figure 3 receives the following representation (Table 9):

Using a set of training reports, a set of rules is formalized for each section in [4]. For example, in the identification section of the CR we have established "person rules" to extract patient entity. In the treatment section, we can extract a list of medication entities with their dosage as properties by applying "medication rules .." This extracted information can be imported into a structured data repository, where they can be queried or used for other applications. Below, we give some simple rules for extracting entities.

- R1: If (Person Trigger and NAM[4] and NAM) then person entity;
- R2:If (NAM and Person Name) then person entity;
- R3:If (NUM and Date) and (left neighbors = «âgé de») then age property;
- R4:If (NAM) and (size ≤2) and (left neighbors = «demeurant à») then city.

[2] http://www.ims.uni-tuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger
[3] Patient names are not real names
[4] NAM: Proper noun; result of morphosyntactic tagging done by the TreeTagger tool

Table 9. Noun phrase representation

Neighbors on the left		***Noun Phrase***		
$wordt_{-2}$	$word_{-1}$	$word_1$	$word_2$	$word_3$
De	La	Patiente	Aissati	Lyla
Part of Speech →		NOM	NAM	NAM
Semantic Information →		Person Trigger	Unknown	Person Name

For example, the first rule (R1) classifies a term as a person if its first word appears in the list of person triggers. Its second and third words are tagged with proper name category (e.g., "patiente Aissati Lyla").

4.2 Boolean modeling of rules

Our motivation to model rules for extracting entities with the Boolean principle adopted by the cellular automaton CASI is to reduce the storage complexity of these rules and also the response time while using them for classification.

We adopt the following approach as shown in Fig. 4. We first transform rules and secondly we produce Boolean rules. Thus, the rules previously described will be pre-processed and transformed, which will generate the Boolean rules.

4.2.1 Coding

We present in this section the coding adopted by the cellular automaton CASI to represent the different information used for named entities extraction. Table 10 provides some variables used by rules to describe NP.

Table 10. Some Descriptive Variables and Coding

Descriptive Variables	Notation	Value
Semantic information of $word_1$	SIW1	= 0 (person trigger) ; = 1(place trigger) ; =2 city) …
Semantic information of $word_2$	SIW2	=0 (name) ; =1(date), …
Part of speech of $word_1$	POS1	=0(NAM) ; =1(NOM) ; =2(NUM) ;
Part of speech of $word_2$	POS2	=0(NAM) ; =1(NOM) ; =2(ADJ)
Part of speech of $word_3$	POS3	=0(NAM) ; =1(ADJ) ;
Term size	TS	>2 ; ≤2 ;
First neighbor-$_1$ on the left	LN1	=0 (« à ») ; =1(« de ») ; =2(« pour ») ;
Second neighbor-$_2$ on the left	LN2	=0(« demeurant ») ; =1(âgé) ; =2(« admis ») ;
Entity class	C	Person, Disease, medication, …

4.2.2 Transformation

The CELFACT and CELRULE layers of CASI represent transformed rules. The input/output matrices represent the input and output relationships between the various facts.

To illustrate this representation, we consider that our knowledge base consists of only two rules, R1 and R3, as cited in Section (§4.1.). According to the coding presented in Table 10 we have the following transformation:

Table 11. Boolean Knowledge base: CELFACT, CELRULE, RE and RS

CELFACT	*EF*	*SF*
SIW1=0	0	0
SIW1=2	0	0
SIW2=1	0	0
POS1=2	0	0
POS2=0	0	0
POS3=0	0	0
LN1=”à”	0	0
LN2=”âgé”	0	0
C=Person	0	0
C=Age	0	0

CELRULRE	*ER*	*SR*
R1	0	1
R3	0	1

$\boldsymbol{R_E}$	*R1*	*R2*
SIW1=0	1	0
SIW1=2	0	0
SIW2=1	0	1
POS1=2	0	1
POS2=0	1	0
POS3=0	1	0
LN1=”de”	0	1
LN2=”âgé”	0	1
C=Person	0	0
C=Age	0	0

$\boldsymbol{R_S}$	*R1*	*R2*
SIW1=0	0	0
SIW1=2	0	0
SIW2=1	0	0
POS1=2	0	0
POS2=0	0	0
POS3=0	0	0
LN1=”de”	0	0
LN2=”âgé”	0	0
C=Person	1	0
C=Age	0	1

- R1: If (SIW1= =0) and (POS2= =0) and (POS3= =0) then C is person
- R3 : If (LN1= = « de ») and (LN2= « âgé ») and (POS1==2) et (SIW2==1) then C is age

This rule set, which is integrated into the cellular automaton CASI, is shown in Table 11 (for this illustration we use only some descriptive variables). To build the Boolean Knowledge base, we use the following principles:

- Every premise or conclusion of a rule is represented by a cell in the CELFACT layer
- *0* initializes the EF state of each cell in CELFACT.
- Every rule in the original base constitutes a cell in the CELRULE layer.
- 0 initializes the ER state of each cell in CELRULE.
- For every fact *f* belonging to *CELFACT* and for every rule *r* in *CELRULE*, if (*f)* is a premise of (*r*) then $R_E[f, r]=1$ otherwise *0*.
- For every fact *f* belonging to *CELFACT* and for every rule *r* in *CELRULE*, if (*f*) is a conclusion of (*r)* then *RS[f, r]=1* otherwise *0*.

4.3 Boolean inference

To explain the principle of this Boolean inference, we consider the knowledge base shown in

Table 12. Boolean inference

(a) Initialization

CELFACT	*EF*	*SF*
SIW1=0	1	0
SIW1=2	0	0
SIW2=1	0	
POS1=2	0	0
POS2=0	1	0
POS3=0	1	0
LN1="à"	0	0
LN2="âgé"	0	0
C=Person	0	0
C=Age	0	0

(b) Rules filtering

CELFACT	*EF*	*SF*
SIW1=0	1	1
SIW1=2	0	0
SIW2=1	0	0
POS1=2	0	0
POS2=0	1	1
POS3=0	1	1
LN1="à"	0	0
LN2="âgé"	0	0
C=Person	0	0
C=Age	0	0

CELRULE	*ER*	*SR*
R1	*1*	*1*
R3	*0*	*1*

(c) Execution of selected rules

CELRULE	*ER*	*SR*
R1	1	0
R3	0	1

CELFACT	*EF*	*SF*
SIW1=0	1	1
SIW1=2	0	0
SIW2=1	0	0
POS1=2	0	0
POS2=0	1	1
POS3=0	1	1
LN1="à"	0	0
LN2="âgé"	0	0
C=Person	1	1
C=Age	0	0

Table 11 and the noun phrase "*patiente Aissati Lyla ..*" We first recall the functioning of the cellular automaton CASI. It consists of three phases: selection, filtering, and execution.

Before running the inference engine *CIE*, we must initialize the *CELFACT* layer with information about the nominal phrase to analyze. In the case of the noun phrase "*patiente Aissati Lyla*" the following cells are set (EF=1). See Table 12(a).

- Patiente (is a person trigger) → EF(SIW1=0)=1
- Aissati (is NAM) →EF(POS2=0)=1
- Lyla (is NAM) → EF(POS3=0)=1

In the filtering phase, the transition function δfact is executed to select only established facts. Those facts can participate in the filtering step. SF (output of facts) receives the initial value of *EF* (input facts).

The *ER* (input rules) is obtained according to the rule $ER = ER + R_E^T \times EF$, as shown in Table 12(b). The cellular engine compares the premises of the rules to all established facts. In our case, the rule *R1* is a candidate and its *ER* receives the value of 1.

During the execution phase, the transition function $\delta\, rule$ is started. At this stage, one or more rules that must be actually called are set. In our case it is only rule *R1* that has been selected and

therefore its output *SR* receives the value 0, as shown in Table 12(c). The conclusion fact of that rule (person) is established in the CELFACT base by *EF=EF + (SR×ER)*.

The cellular automaton implemented for entity classification performs only one cycle after each selection step. As shown in Table 12(c), entries of cells "*SIW1= 0* ", "*POS2 = 0* " and "*POS3 = 0*" are active (their *EF = 1*). After one cycle of inference, the cellular automaton determines the class of "Aissati Lyla" as a person and the *EF* cell of "*C=Person*" fact becomes active (*EF=1*).

5. EXPERIMENTATION AND DISCUSSION

In this section we describe the data and metrics used to test our approach experimentally and discuss the different results.

5.1 Data: Clinical Reports

We have analyzed 50 clinical reports to build the knowledge base, and for the test we created the data set from 15 new clinical reports from patients seen in 2009/2010 at the infectious disease department of the University Hospital of Oran (Algeria). With the help of a doctor, we annotated the test data. We identified 348 different entities. For each of these we recorded the name (person, disease, symptom, etc.) Fig. 5 summarizes the various entities present in the test data.

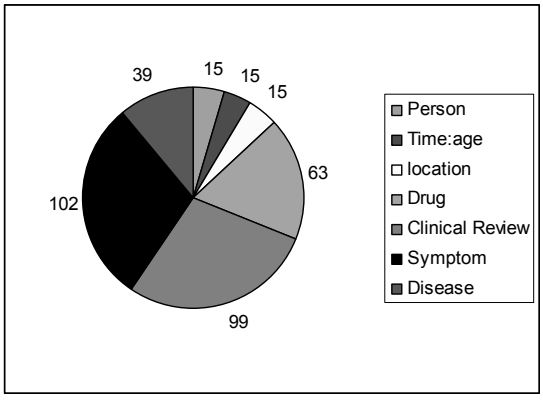

Fig. 5. The different entities present in the test data

5.2 Metrics

These are standard metrics for evaluating named-entity detection. Three metrics were used to measure the accuracy of named-entity detectors: Precision, Recall, and F-measure.They are defined as:

$$Precision = \frac{NEC}{NEC + NEM} \; ; \quad Recall = \frac{NEC}{NEC + NEN} \; ; \quad F - mesure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Where

- NEC: number of named entities identified correctly (True Positives);
- NEM: number of noun phrases mistakenly claimed to be entities (False Positives);
- NEN: number of entities not identified (False Negatives).

5.3 Experimental results

Fig. 6 shows the precision, recall, and F-measure for each class. Analysis of the results of this experiment allowed us to better understand the reasons for the decline in performance, especially recall, that were relative to certain entities. For example, the low recall was due to the insufficient coverage of the diversity of disease, symptoms, and findings expressions in our small set of rules. The system fails to recognize entities because it does not have rules for catching these. This case was especially apparent for entities like symptoms, and clinical review.

Globally, the system performs a good extraction. Out of 348 entities, it accurately matched 278, missed 70 (false negative), and identified 28 entities erroneously (false positive). This gives a precision of 92% (macro precision) and a recall of 89% (macro recall). These results are very interesting but need to be checked in a collection of clinical reports, which is more important. The Cellular automaton relies on a library of rules and a lexicon of proper nouns to identify entities. Fortunately, both the lexicon and the rules are flexible, and can be easily customized to better extract the missing named entities.

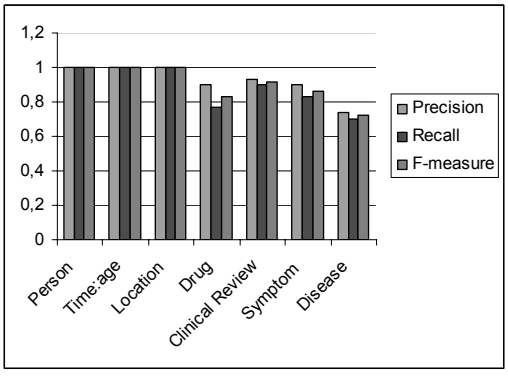

Fig. 6. Performance system

6. CONCLUSION

In this paper, we proposed the passage of a classic named entities classification to a cellular classification. In fact, we have explicitly shown the functioning of the cellular automaton for named entities classification.

Our motivation to adopt the principle of cellular automata for this classification task is to exploit its advantages which are knowledge representation and computation time during classification. The Boolean principle offers the following advantages:

- In the form of binary matrices, the representation of knowledge and its controls are simple

and require minimal pretreatment.
- Ease of implementation of the transition functions, which are robust and have a low complexity.
- Determining the class for an entity results from the execution of a single cycle of the cellular engine.

This new Boolean representation will reduce the amount of storage and execution time. In fact, this is due to the use of Boolean representation matrices RE and RS, and the multiplication used by Boolean transition functions: *δfact* and *δrule*.

Boolean matrices can be expressed as two vectors of several binary sequences and the amount of memory required to store these Boolean matrices is in the order of O (q), when using the q sequences of the r-bit.

The proposed approach has the ability to take into account an initial knowledge base in the form of symbolic rules and convert it according to the principle of Boolean cellular automaton CASI. In the context of improving the performance of this new solution, we plan to add knowledge (rules) by machine learning. The COG component of the automaton CASI, which is based on supervised learning, allows us to make the induction from data. This hybridization will be a performance evaluation for future work. We therefore propose as a perspective or increasing the rate of named entities extraction by using the automatic acquisition of Boolean classification rules. What we are going to baptize a named entities extraction tool that is guided by data mining.

REFERENCES

[1] A. R. Aronson, "*Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program*", *American Medical Informatics Association Annual Symposium*, AMIA'01, Washington, DC, USA, 2001, pp.17-21.

[2] B. Atmani, B. Beldjilali, "*Knowledge discovery in database: Induction graph and cellular automaton*", Computing and informatics journal, Vol.26, 2007, pp.171-197.

[3] F. Barigou , B. Atmani , M. Mokaddem , B. Beldjilali, "*Towards an Automated system for extracting named entities from medical reports*", Premier congrès international sur les modèles, optimisation et sécurité des systèmes, Tiaret, Algeria, 2010.

[4] F. Barigou, B. Beldjilali, B. Atmani, "*MEDIX: Medical Information eXtraction from clinical Reports." International Conference* on Communication, Computing and Control Application, Hammamet, Tunisia, March 3-5, 2011, pp.488-494

[5] A. Castilla, S. Furuie, E. Mendonca, "*Multilingual information retrieval in thoracic radiology: feasibility study*", Medinfo, 2007, pp.387-91.

[6] W.Chapman, J. Dowling, M. Wagner, "Fever Detection from Free-text Clinical Records for Biosurveillance", *Journal of Biomedical Informatics*, Vol.37, No.2, 2004, pp.120-127.

[7] M. Chau, J. Xu, H. Chen, "*Extracting Meaningful Entities from Police Narrative Reports*", Proceeding of the *National Conference for Digital Government Research*, 2002, pp.271-275.

[8] D. Chhieng, T. Day, G. Gordon, J. Hicks, "*Use of natural language programming to extract medication from unstructured electronic medical records*", *American Medical Informatics Association Annual Symposium,* AMIA'07.

[9] C. Clark, K. Good, L. Jezierny, M. Macpherson, B. Wilson, U. Chajewska, "*Identifying smokers with a medical extraction system*", *American Medical Informatics Association Annual Symposium*, AMIA'08, 2008.

[10] M. Embarek, O. Ferret, "*Learning patterns for building resources about semantic relations in the*

medical domain", Proceedings of the International Conference on Language Resources and Evaluation, LREC'08, Marrakech, Morocco, 26 May - 1 June, 2008.

[11] C. Friedman, G. Hripcsak, "Evaluating natural language processors in the clinical domain". *Methods of information in Medicine*, 1998, Vol.37, pp.334-344.

[12] C. Friedman, P. Alderson, J. Austin, J. Cimino, S. Johnson*, "A general natural language text processor for clinical radiology*", Journal of the *American Medical Informatics Association*, 1994, Vol.1, No.2, pp.161-174.

[13] P. Haug, L. Christensen, M. Gundersen, B. Clemons, S. Koehler, K. Bauer, "*A natural language parsing system for encoding admitting diagnoses*", American Medical Informatics Association Annual Symposium, AMIA 97, 1997, pp.814-818.

[14] N. Jain, C. Friedman, "*Identification of Findings Suspicious for Breast Cancer Based on Natural Language Processing of Mammogram Reports*", Proceedings of the *Fall AMIA Conference*, Philadelphia, USA, 1997, pp.829-833.

[15] J. Mork, O. Bodenreider, D. Demner-Fushman, R. Doğan, F. M. Lang, "Extracting Rx information from clinical narrative", Journ*al of the American Medical Informatics Association*, JAMIA 2010, Vol.17, No.5, pp.536-539.

[16] M. Jiang, Y. Chen, M. Liu, T. Rosenbloom, S. Mani, J. Denny, H. Xu, "*A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries*", Journal of *the American Medical Informatics Association* , JAMIA 2011;Published Online First: 20 April 2011 doi:10.1136/amiajnl-2011-000163.

[17] C. A. Knirsch,, N. Jain, A. Pablos-Mendez, C. Friedman, G. Hripcsak, "Respiratory Isolation of Tuberculosis Patients Using Clinical Guidelines and an Automated Clinical Decision Support System" , *Journal Infection Control and Hospital Epidemiology*, 1999, Vol.19, No.2, pp.94-100.

[18] M. Levin, M. Krol, A. Doshi, D. Reich, "*Extraction and mapping of drug names from free text to a standardized nomenclature*", AMIA, Annual Symposium Proceeding, 2007, pp.438-442.

[19] S. Meystre, G. Savova, K. Kipper-Schuler, J. Hurdle, "*Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research*", Yearbook of Medical Informatics. 2008, pp.128-44.

[20] N. Nadeau, S. Sekine, "a survey of named entity recognition and classification", Journal of *Linguistic Investigations*, 2007, Vol.30, No.1, pp.3-26.

[21] T. Poibeau, "Boosting the robustness of a named entity recognizer", International Journal of Semantic Computing, 2009, Vol.32, No.1, pp.77-98.

[22] A. Roberts, R. Gaizauskas, M. Hepple, N. Davis, G. Demetriou, Y. Guo, J. Kola, I. Roberts, A. Setzer, A. Tapuria, B. Wheeldin, "The CLEF corpus: semantic annotation of clinical text", AMIA Annual Symposium proceedings Volume: 2007, Publisher: *American Medical Informatics Association*, pp.625-629.

[23] G. Shadow, C. MacDonald, "*Extracting structured information from free text pathology reports*", AMIA Annual Symposium Proceeding, Washington, DC, 2003.

[24] T. Sibanda, T. He, P. Szolovits, O. Uzuner, "*Syntactically-informed semantic category recognition in discharge summaries*", Proceedings of the Fall Symposium of the American Medical Informatics Association; Washington, DC, November, 2006.

[25] O. Uzuner, I. Goldstein, Y. Luo, I. Kohane, "*Identifying Patient Smoking Status from Medical Discharge Records*", Journal of the American Medical Informatics Association, January 2008, Vol.15, No.1, pp.14-24.

[26] Y. Wang, "*Annotating and Recognising Named Entities in Clinical Notes*", Proceeding of the ACL-IJCNLP 2009 Student Research Workshop, Singapore, 2009, pp.18-26.

[27] H. Xu, S. Stenner, S. Doan, K. Johnson, L. Waitman, J. Denny, "*MedEx: a medication information extraction system for clinical narratives*", Journal of American Medical Informatics Association, 2010, Vol.17, No.1, pp.19-24.

[28] H. Yang, I. Spasic, J. Keane, G. Nenadic, "A Text Mining Approach to the Prediction of a Disease Status from Clinical Discharge Summaries", Journal of *the American Medical Informatics Association*, 2009, Vol.16, No.4, pp.596-600.

Fatiha Barigou

She is a computer science teacher in the Department of Computer Science of Oran University (Algeria). She earned her Master of Science degree in 1998 from Oran University. She is currently a Ph.D. candidate in the Computer Science Department at the same university. Her research interests focus on text mining, information extraction, and information retrieval areas.

Baghdad Atmani

He received his Master of Science degree in 1996 from the Department of Computer Science in Oran (Algeria). He is currently a Ph.D. in the Computer Science Department at the University of Oran. His research interests include knowledge discovery in databases, data mining, feature selection, neural networks, and cellular automata.

Bouziane Beldjilali

He received his Ph.D. degree in Computer Science from the University of Oran (Algeria), in 1996. He is a professor in the Computer Science Department at the University of Oran. His research interests include formal specifications, knowledge management, databases, artificial intelligence, and automatic learning.