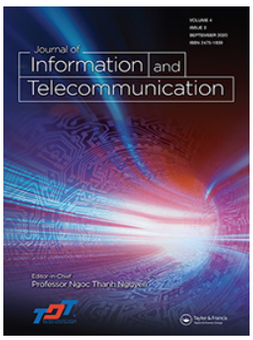

Journal of Information and Telecommunication

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjit20

Event detection based on open information extraction and ontology

Sihem Sahnoun , Samir Elloumi & Sadok Ben Yahia

To cite this article: Sihem Sahnoun , Samir Elloumi & Sadok Ben Yahia (2020) Event detection based on open information extraction and ontology, Journal of Information and Telecommunication, 4:3, 383-403, DOI: 10.1080/24751839.2020.1763007

To link to this article: https://doi.org/10.1080/24751839.2020.1763007

Published online: 12 May 2020.

JOURNAL OF INFORMATION AND TELECOMMUNICATION
2020, VOL. 4, NO. 3, 383–403
https://doi.org/10.1080/24751839.2020.1763007

Event detection based on open information extraction and ontology

Sihem Sahnoun[a], Samir Elloumi [a,b] and Sadok Ben Yahia [a,c]

[a]University of Tunis El Manar, Faculty of Sciences of Tunis, LIPAH-LR11ES14, Tunis, Tunisia; [b]Information Systems Department, Faculty of Computing and Information Technology in Khulais, University of Jeddah, Jeddah, Saudi Arabia; [c]Department of Software Science, Tallinn University of Technology, Tallinn, Estonia

ABSTRACT

Most of the information is available in the form of unstructured textual documents due to the growth of information sources (the Web for example). In this respect, to extract a set of events from texts written in natural language in the management change event, we have been introduced an open information extraction (OIE) system. For instance, in the management change event, a PERSON might be either the new coming person to the company or the leaving one. As a result, the Adaptive CRF approach (A-CRF) has shown good performance results. However, it requires a lot of expert intervention during the construction of classifiers, which is time consuming. To palpate such a downside, we introduce an approach that reduces the expert intervention during the relation extraction. Also, the named entity recognition and the reasoning, which are automatic and based on techniques of adaptation and correspondence, were implemented. Carried out experiments show the encouraging results of the main approaches of the literature.

ARTICLE HISTORY
Received 14 January 2020
Accepted 28 April 2020

KEYWORDS
Information extraction; event recognition; named entity; relationship; OIE; ontology

1. Introduction

Information Extraction (IE) is one of the hottest areas of the active research in artificial intelligence. It was developed in the late 1980's and 1990's with the Message Understanding Conferences (MUC) (Grishman & Sundheim, 1996), in the latter, a set of evaluation campaigns have been suggested and have defined the different tasks of the IE systems. The core task of IE is named entity recognition (NER), which is based on the extraction of categorizable textual objects in classes such as names of people, names of organizations, etc.

The relation identification is also a worthy of mention task in the field of IE, which aims to find the mention of a binary relation between two entities in a text (Elloumi et al., 2013). Another specific type of knowledge that can be extracted from the text is the event and can be considered as an object that admits an existence in the time space and depends on other objects (Elloumi et al., 2013). The event extraction is domain dependent. It has been also studied for more than two decades through the MUC and the Automatic Content Extraction (ACE) programs. Each year of the MUC program focussed on a single type of

CONTACT Samir Elloumi samir.elloumi@fst.utm.tn

event template, allowing for the study of complex structure and fillers within the event. Among the types of events studied in the MUC, Fleet operations, terrorist activities in Latin America, corporate joint ventures, microelectronic production, etc.

With the ACE program, researchers have switched focus to more general types of events, such as conflicts, transportation of people/items, and life events. This allows the program to capture a much wider range of event types than the MUC program but still remains domain dependant and requires expert's involvement.

In 2007, the open information extraction (OIE) has appeared and has allowed the task of extracting knowledge from texts without much supervision. OIE systems aim to obtain relation tuples with a highly scalable extraction by identifying a variety of relation phrases and their arguments in arbitrary sentences (Vo & Bagheri, 2017).

In this paper, which is an extended version of our ICCCI 2019 conference paper, we propose an event extraction approach which reduces the expert intervention by using an OIE system for a relation extraction without supervision, an automatic NER, and an ontology applied for any management change event. We show some case application scenarios for a management change event domain followed by experimental result in terms of recognition rates.

The remainder of this paper is organized as follows. In Section 2, we present the related works of the IE domain and its different levels. Section 3 thoroughly describes our new approach for IE. In Section 4, we present the experimental results we obtained for the management change event. In Section 5, we conclude and sketch issues of future work.

2. Related work

According to the type of information extracted, different levels are defined in the domain of IE. In the following section we provide a general overview of these levels and elaborate further in subsequent sections.

2.1. Named entity recognition

A named entity (NE) is often a word or an expression that represents a specific object of the real world. The NEs generally correspond to the names of the person, organization, place and dates, monetary units, percentages, units of measurement, etc.

Therefore NER is defined as an important task among the tasks of IE, which has been satisfactorily carried out for well-formed texts such as news story for certain languages like english and french. NER consists of searching and identifying objects which are categorizable into two classes (Satoshi et al., 2002):

- *Named entities* can be the name of people, name of organizations, name of places, etc. The name of people extractor aims to identify the first and the last name, the places extractor based on a list of files that contains the name of places and their types (cities, countries, governorate) and the organization extractor aims to identify the name of organization and its type (Association, society, Faculty, etc).
- *Numeric entities* can be temporal expressions and numbers. The temporal expressions can be dates and any other temporal markers (period, date, etc). Numbers can be numerical expressions such as money and percentages.

There are several NER systems that exist and are classified into three broad families: Systems based on *symbolic approaches*, systems based on *learning approaches* and *hybrid systems* (Talha et al., 2014).

The symbolic approach is based on the use of formal grammars built by hand and exploit syntactic labelling associated with words, such as the grammatical category of the word. It is also based on dictionaries of proper names, which usually include a list of the most common names, first names, place names, and names of organizations.

Shaalan and Raza (2009) developed their NERA system to extract ten types of NEs. This system relies on the use of a set of NE dictionaries and a grammar in the form of regular expressions.

To learn patterns that will recognize entities, *learning-based methods* use annotated data. The annotated data corresponds to documents in which the entities, with their types, are indicated. Subsequently, a learning algorithm will automatically develop a knowledge base using several numerical models such as Conditional Random Fields (CRF), Support Vector Machines (SVM), Hidden Markov Model (HMM), etc., which is not the case for symbolic approaches that only apply the previously injected rules. Benajiba et al. (2009) developed an SVM learning technique for their NER system using a set of features of the Arabic language.

The combination of the two antecedent approaches represent the emergence of an *hybrid approach*. It uses rules written manually but also builds some of its rules based on syntactic information and information extracted from learning data. Abuleil (2006) has adopted an hybrid approach for extraction of entities in arabic, taking advantage of symbolic and learning approaches.

As the recent advancement in the deep learning (DL), it produces huge differences in accuracy compared to traditional methods for Natural Language Processing (NLP) tasks. Entity extraction from text is a major Natural NLP task. According to Chiu and Nichols (2016), the implementation has a bidirectional long short-term memory(BLSTM) at its core and a convolutional neural network (CNN) to identify character-level patterns.

2.2. Relation extraction

Relation extraction (RE) is an important task for many applications and many studies. The RE task involves identifying relationships between named entities in each sentence of a given document. A relation usually indicates a well-defined relation (having a specific meaning) between two or more NEs.

The existing IE systems can be roughly categorized along two dimensions: The supervision required and the used models.

A system can be fully supervised, semi-supervised, or distantly supervised. The second dimension is that the model used can be pattern-based, not pattern-based (most of them are feature-based sequence classifiers), or an hybrid of pattern and feature-based.

- *The pattern-based fully supervised approach*
  Systems differ in how they create patterns, learn patterns, and learn the entities they extract. It has been very successful at extracting richer relations using rules and NEs. The work by Hearst (1992) used hand written rules to automatically generate more rules that were manually evaluated to extract hypernym-hyponym pairs from text.

Berland and Charniak (1999) used patterns to find all sentences in a corpus containing basement and building.

- *The pattern-based distantly supervised*
  The matching of seed sets to text take word ambiguity into account when two different objects with the same lexicalization express two different relations by generating a training data automatically. For example, the DIPRE algorithm by Brin (1998) used string-based regular expressions in order to recognize relations such as author-book, while the SNOWBALL algorithm by Agichtein and Gravano (2000) learned similar regular expression patterns over words and named entity tags.
- *Distantly supervised Non-pattern-based systems*
  Many systems, such as CRFs-based systems, use a set of entities, dictionaries as features. Sarawagi and Cohen (2004) worked on improving the matching of named entity segments to dictionaries and to use it as a feature for a sequence model.
- *Distantly supervised hybrid systems*
  Individual components of a pattern-based learning system can use feature-based learning methods to learn good pattern and entity ranking functions. Niu (2012) has shown promising results on distantly supervised relation extraction of Angeli et al. (2014) by using fast inference in Markov logic networks. Angeli et al. (2014) used learned dependency patterns.
- *Deep learning models for relation extraction.*
  The word embeddings (Mikolov et al., 2013) express each word as a vector and aim to capture the syntactic and semantic information about the word. They are learnt using unsupervised methods over large unlabelled text corpora which were adapted as standard in all subsequent RE deep learning models. Nguyen and Grishman (2015), explore CNNs for Relation Extraction and Relation Classification tasks, the model completely gets rid of exterior lexical features to enrich the representation of the input sentence and lets the CNN learn the required features itself.
- *Open information extraction*
  OIE, a popular task in recent years, it extracts relations from the web with no training data and no list of relations. ReVerb (Fader et al., 2011) and OLLIE (Durme & Schubert, 2008) learn how to extract open-domain relation triples (c.f., Subsection 3.2.1 describes how we used OLLIE in our system).

2.3. Event recognition

Event extraction is intended to extract from the text a characterization of an event, defined by a set of entities associated with a specific role in the event.

Some techniques use *data-driven approaches*, others use *knowledge-driven approaches* and others use *hybrid approaches*.

- *Data-driven approaches* require a large text corpora in order to develop models that approximate linguistic phenomena. Data-driven methods require a lot of data and a little domain knowledge and expertize, while having a low interpretability. They don't deal with meaning explicitly, i.e. they discover relations in corpora without considering semantics.

For example, Conditional Random Field based (CRF) (Sarawagi & Cohen, 2004) systems apply the classifier to a set of texts to produce a set of annotated texts. The interest and efficiency of CRFs come from taking into account the dependencies between labels related to each other in the graph.

- *knowledge-driven approaches* is often based on models that express rules representing expert knowledge. It is intrinsically based on linguistic and lexicographical knowledge, as well as on existing human knowledge concerning the content of the text to be treated. This alleviates the problems with the statistical methods concerning the meaning of the text. For example, the GLAEE approach (Elloumi et al., 2013) is based on the generation of annotation patterns that involves a list of keywords and cue words. Doing so, we identify events in the learning phase, afterwards the annotation phase is performed by an alignment between the pattern and the new text.
- *hybrid approaches* seem to be a compromise between data and knowledge-based approaches, requiring an average amount of data and domain knowledge and having medium interpretability. However, it should be noted that the amount of expertize required is high, due to the fact that several techniques are combined. For example, the interest of Two-level approach (Elloumi, 2019) is to adapt the recognition of named entities level to the CRF tool based on learning techniques and a correspondence between level 1 learning (PERSON, ORG, DATE, NUMEX, PROFIL) as well as learning level 2 (NEW PERSON, COMING PERSON), which brings us back to a double generation of the classifier.

Table 1 summarizes the different approches. The diversity of approaches for IE witnesses the steady interest and usefulness of this domain. Several systems have been proposed in the context of event extraction. We have studied their functionality and, according to their limits, we have introduced our new approach. GLAEE (Elloumi et al., 2013) is an approach based on the generation of annotation patterns which involves a list of keywords and cue words. It purposes to identify events by an alignment between the pattern and the new text. Here as stated before, the entities, the keywords and the cue words require human intervention.

The template filling through information extraction (Kodelja et al., 2017) is an approach that aims to classify sentences according to a predefined event type. The event classification is then generally assimilated to the detection of triggers within the sentence. The trigger type indicates the type of the template form. Here the rules are written

Table 1. Summarizing table.

Approach	Language	Method	Advantage	Disadvantage
GLAEE	English	Knowledge-driven approach	The average recognition rate is 68.93%	The keywords are defined by the expert
CRF	All the languages	Data-driven approach	Gives a probability to each returned extraction is an index of quality of the information	Difficult to access and to be modified
Template filling through IE	French	Knowledge-driven approach	Identifying multiple events within a sentence	The rules are written manually
Two-level approach for extracting events	English	Data-driven approach	The average recognition rate is 67.91%	Depends on the quantity and the quality of the learning data

manually and require a lot of data. The A-CRF system uses CRFs (Sarawagi & Cohen, 2004) which are based on a learning phase to manually prepare a set of corpora to train the model and an annotation phase to annotate a new text based on the learned texts and the probability distribution. Hence, the learning phase was applied to generate the level 1 classifier. The corpus is also prepared for deriving the level 2 classifier after an adaptive phase. It means replacing all annotated texts by their corresponding NEs.

We present below a summarizing table that glances all the work on event extraction from the approaches we studied. This study was carried out according to the methods used for event extraction, the language applied as well as the advantages and disadvantages of each of the approaches.

In general, the event extraction task is a dependent domain. It requires human intervention in order to construct manually the annotation rules or to prepare an annotated corpus as an input for the learning phase. In order to reduce the expert intervention, we suggest using ontologies as a knowledge source for describing any event. We suppose that an event-ontology describes the relations between named entities and their possible roles in an event. For instance, in the management change event, a person role might be the new-person, i.e. the hired one or the leaving person. To check whether a new text informs about a given event, in particular the roles of some named entities, a matching process is required between the new text and the ontology. To do that, we suggested the following steps : First, we applied the OIE in order to extract the most relevant relations within a text, then we applied the NER on these relations. And finally, we made a matching between results of the previous step and the ontology for deriving a possible event.

3. Our approach: event detection based on open information extraction and ontology

The main interest of our approach is how to extract a specific information from all existing relationships between all entities that can be found in a text.

Our goal is to reduce human intervention during event extraction. In particular, we present our approach which is based on two phases that depend on each other as shown in Figure 1. The first phase is the modelling of an event by an ontology and constructing a set of rules acquired manually. The second is the recognition phase which includes the RE, the NER and an automatic reasoning between learning rules and an input ontology adaptation. Our aim through these two phases is an efficient event extraction.

3.1. Learning phase

An event is an object that admits an existence in the space of time and depends on other objects in relation. To cope with such constructs, we build an ontology manually to model these events in any domain. An ontology is a set of concepts, as well as relationships between these concepts[1]. Algorithm 1 describes the mechanism of the learning phase on a list of named entities, relationships and roles to produce as an output, an ontology and a set of rules. In our case, the concepts stand for the named entities of the event (Person, Organization, etc.) and their roles (Coming_person, Leaving_person, IN_ORG) being in certain relationships.

Figures 2 and 3 illustrate respectively the learning phase which is composed of event ontology (classes, subclasses and relations) and rules construction. In particular, we have:

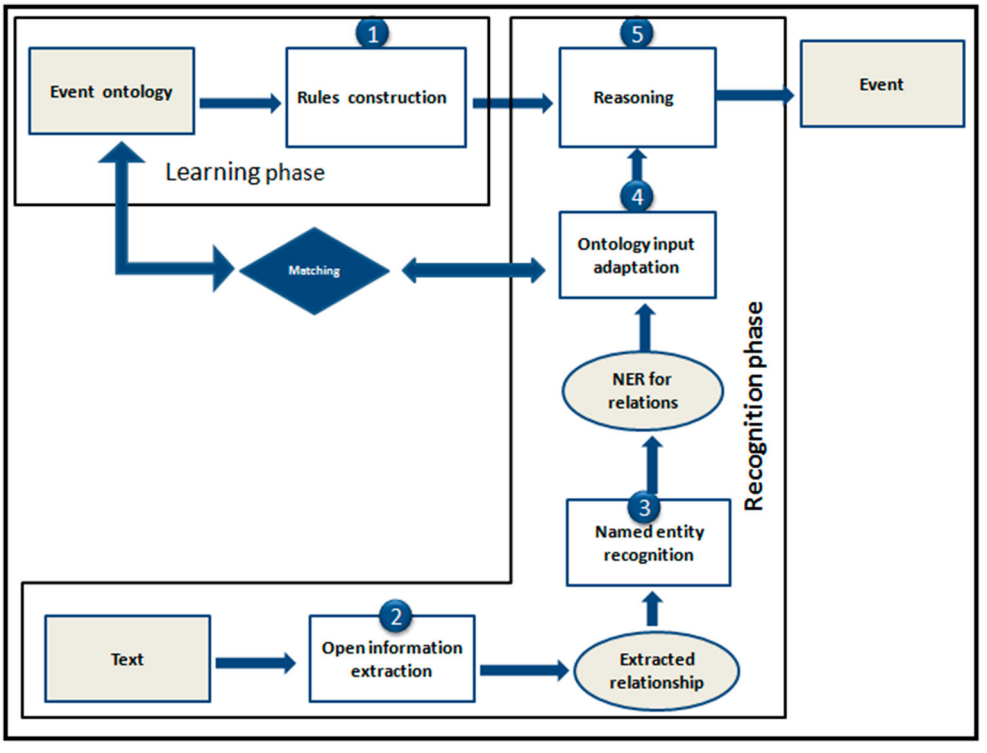

Figure 1. The overall architecture of the introduced approach at a glance.

- The class Person has two subclasses: Coming_person and Leaving_person.
- The class Organization has two subclasses: IN_ORG and OUT_ORG.
- The class Position has four subclasses : CP_new_position, CP_previous_position, LP_previous_position, LP_new_position.
- The class Date has two subclasses : Date_of_coming, Date of leaving.

Algorithm 1 Learning phase

Input
L1 : List of entities
L2 : List of roles
R : List of relations
Output
O : Ontology
RC : Rules Construction
foreach Entity i in L1 **do**
Class ← *Class* ∪ *add*(*i*) *//Add named entities as classes in the ontology*
foreach Role j in L2 **do**
Subclass ← *Subclass* ∪ *add*(*i*, *Class*) *// Add roles as subclasses for named entities that represent classes in ontology*
end
end
foreach RelationrinR **do**
Relation ← *Relation* ∪ *add*(*r*) *//Add a set of relationships to the ontology*
end
O ← *Event_Ontology*(*Class*, *Subclass*, *Relation*) *//The event is modeled by these classes, subclasses and relationships*
RC ← *Rules_Construction*(*Class*, *Subclass*, *Relation*) *//The rules are made by these classes, subclasses and by relationships*

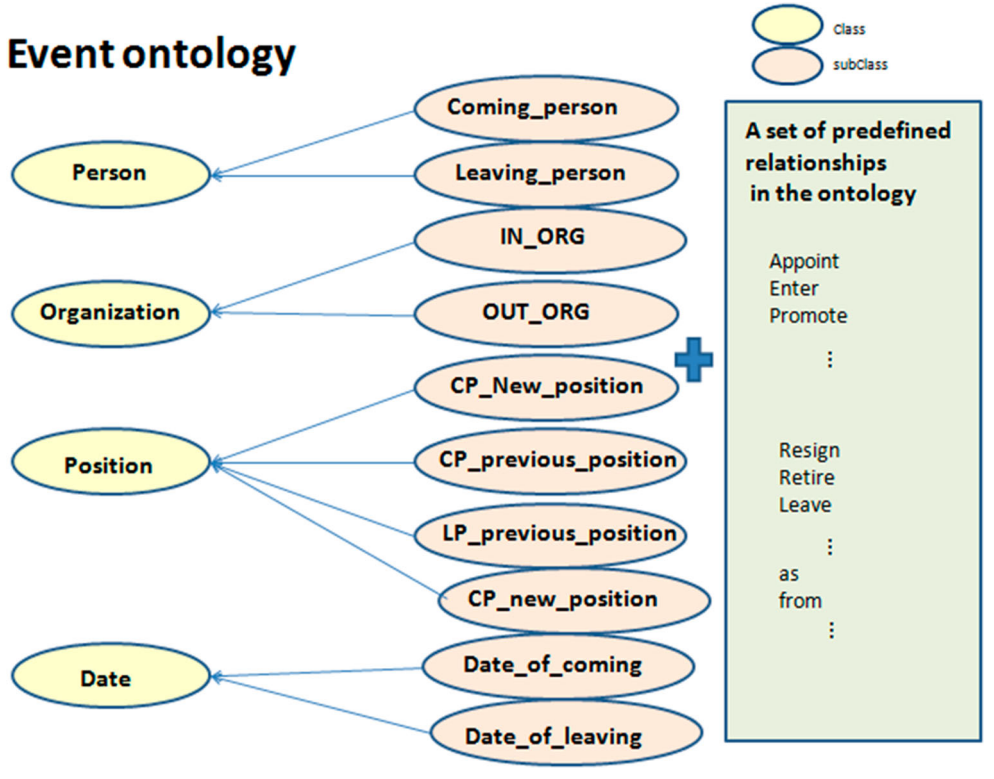

Figure 2. Example of an Event ontology.

```
Person(?x) ^ appoint(?o, ?x) ^ Organization(?o) -> IN_ORG(?o) ^ Coming_person(?x)
Position(?p) ^ as(?x, ?p) ^ Coming_person(?x) -> CP_new_position(?p)
Position(?p) ^ as(?x, ?p) ^ Leaving_person(?x) -> LP_previous_position(?p)
Position(?p) ^leave(?x, ?p) ^ Leaving_person(?x) -> LP_previous_position(?p)
Person(?x) ^ resign(?o, ?x) ^ Organization(?o) -> OUT_ORG(?o) ^ Leaving_person(?x)
```

Figure 3. Example of the built rules.

Our ontology contains a set of predefined relationships to connect two eventual instances such as appoint, resign, promote, etc.

Informally, a rule may be read as meaning that if the antecedent holds (is 'true'), then the consequent must also hold. The rules construction is an important step in our approach which drives us, through the ontology and the result of the recognition phase to a possible event extraction. A set of rules are predefined to assign to every instance its role.

For instance the following rule:

Person(?x) $\wedge$ *appoint(?o,?x)* $\wedge$ *Organization(?o)*$\rightarrow$ *IN_ORG(?o)* $\wedge$ *Coming_Person(?x)* means that any instance 'x' of type Person is connected by an 'appoint' relation with any instance of type Organization 'o' gives us the result that the instance 'x' is a Coming_person and the instance 'o' is an IN_ORG. For the learning phase, we chose protég as an ontology

modelling tool[2]. For building this rule set, we used Semantic Web Rule Language (SWRL) tab. The later is a Protégé plugin that provides a development environment for working with SWRL rules.

3.2. Recognition phase

The recognition phase, as described by Algorithm 2, operates through four steps which are thoroughly explained in the remainder.

3.2.1. Step 1: open information extraction

The input of the system is a text in natural language, the first step of recognition is generated by an OIE system which allows to extract textual relationship triplets existing in each sentence. The relationship triplets contain three textual components (Arg1, Rel, Arg2) where the first and the third, respectively, stand for the pair of arguments and the second indicates the relationship between them. However, the second OIE type handles triplets with attribution and conditions if they exist. It is in the form of *(Arg1; verb; Arg2) [attribution/condition]*. The aim of this step is to restrict the content of the text into relationships that are well defined and have a specific meaning for each sentence in the text. Through Example 3.1, we provide some illustrations of OLLIE extraction process.

The OLLIE tool produces a strong performance by extracting relationships not only mediated by verbs, but also mediated by nouns and adjectives[3]. It can capture N-ary extractions where the relation phrase only differs by the preposition. In addition, OLLIE captures enabling conditions and attributions if they exist.

Example 3.1 Here, we provide some examples of OLLIE extraction process modelling different outputs types.

- *Enabling conditions*
  Sentence: If I slept past noon, I'd be late for work.
  Extraction: (I; 'd be late for; work)[enabler=If I slept past noon]

Algorithm 2 Recognition phase

```
Input
    T : Text
    RC : Rules Construction
    RT : Relation Triplet
Output
    E : Events
Begin
    RT ← OIE(T) //Relation extraction by an open information extraction tool
    R ← RT.substring(pos(';', RT) + 1, − pos(';', RT) − 1)
//Extract the verbal part of the relationships
foreach Relation_triplet rt in RT do
    Token ← Tokenization(rt) //Cut the relationship triplet into tokens
  foreach Token t inrt do
        NE ← NER(t) //Named entity recognition automatically
        A ← Adaptation(NE, R, O)
        RS ← Reasoning(A, RC)
        E ← Event_Extraction(RS) // Event extraction By a reasoner
        end
end
```

- *Attribution*
  Sentence: Some people say Barack Obama was not born in the United States.
  Extraction: (Barack Obama; was not born in; the United States)[attrib=Some people say].
- *N-ary Extraction*
  Sentence: I learned that the 2012 Sasquatch music festival is scheduled for May 25th until May 28th.
  Extraction: (the 2012 Sasquatch music festival; is scheduled for; May 25th)
  Extraction: (the 2012 Sasquatch music festival; is scheduled until; May 28th)
  N-ary: (the 2012 Sasquatch music festival; is scheduled; [for May 25th; to May 28th])

3.2.2. Step 2: named entity recognition

The input of the NER tool is a triplet found in the previous step. The triplet will have an automatic recognition of named entities after a tokenization step. Hence, the system can detect person, organization, location, etc., in any part of the triplet. We used Python as one of the languages commonly used for Automatic Language Processing. Its particularity is to be open source and to solve common problems and especially its 'libraries' which contain different algorithms directly usable by data scientists. Among these libraries we can mention spaCy and nltk[4]. We chose spaCy for NER and NLTK for lemmatisation and tokenization.

As illustration, we present in Figures 4 and 5 two cases of NER. In Figure 4, after recognizing QNB as an Organization, Mark as a person and president as a position the first condition is checked: The number of named entities is greater than or equal to 2 and the part between the delimiters ';' presents the part that contains the relations. In this example, the verb 'appoints' after being lemmatized and the relation 'as' are compared to the list of relations in the ontology.

In Figure 5, the number of entities is greater than 2 and can be linked by a relationship. So input1, input2 and input3 should be entered to the ontology as instances under the classes Person, Position and Organization. In the case of there are triplets that share the same verbal part, then we work on the triplet which has the highest degree of certainty *d*.

3.2.3. Step 3: ontology input adaptation

After recognizing the NEs, the verbs will be passed through a lemmatization layer that will convert conjugated verbs to their infinitive form. Every token recognized by a NE will be added as instance in the ontology. Algorithm 3 illustrates the adaptation step.

During the learning phase, we said that the concepts represent the NEs of the event and their roles. In this step, we operate as follows: Tokens are added as instances to the ontology and will be linked by relations whenever the following conditions are fulfilled :

- The number of named entities is greater than or equal to 2 to have a possible relationship among them. The NE can be found in the triplet or in the attribution/condition part (if it exists) as shown in Figures 6 and 7.
- The lemmatized verb and the other relations between delimiters ';' should be included in the relation list of the ontology and NEs can be linked with these relations. As shown in Figures 8 and 9, respectively, the relations appoint and resign were added to the ontolgy and the reasonning stage could be started.

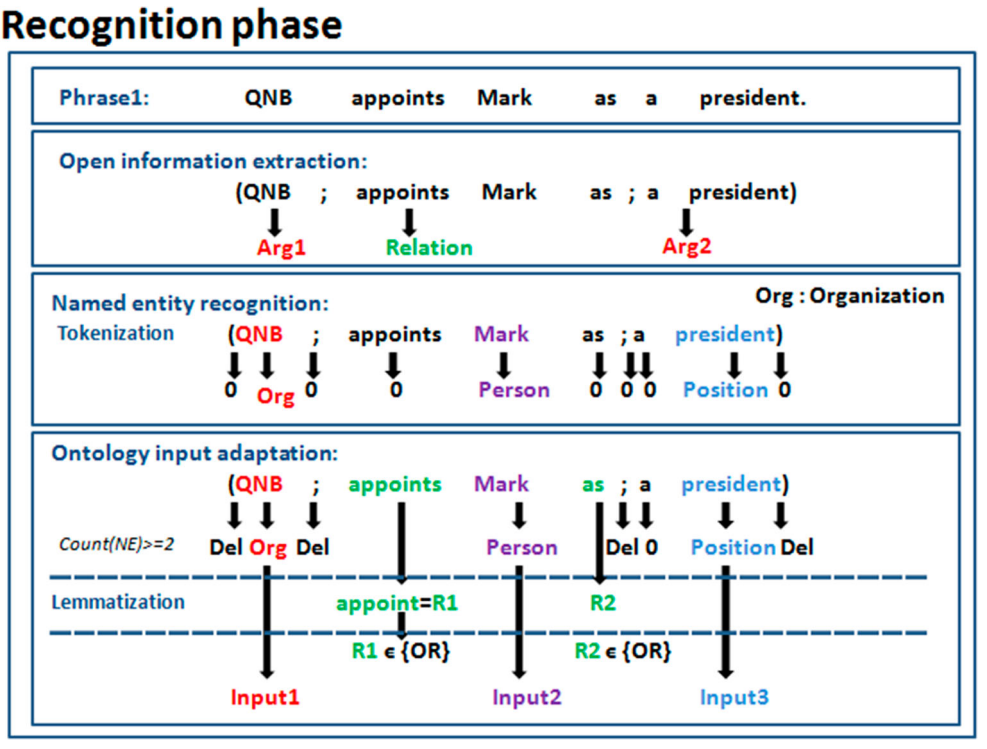

Figure 4. Example of recognition phase for the phrase 'QNB appoints Mark as a president'.

To automatically add the instances (individuals) to the ontology and to link them by their relationship, we used OWLAPI in a Java code for manipulating the elements that compose an ontology (i.e. classes, individuals, properties, annotations, restrictions, etc.) to the ontology and Owlready2 for the reasoning step as a module for ontology-oriented programming in Python. It can load OWL 2.0 ontologies as objects, edit them, save them, and perform reasoning via the HERMIT reasoner (included).

Algorithm 3 Adaptation

Input
NE : Named entity
R : Relation
O : Event_Ontology
Output
tk : tokens_input_ontology
RL : Relation_lemmatizated
Begin
if (triplet.Count(NE)$>=2$) **then**
tk ← *NE.token // tk takes tokens that are recognized by named entities*
if (R in O.Relations) **then**
RL ← *R.Lemmatization // Verbs are transformed into infinitive*
Matching(*tk*, *RL*, *O*)
end
end
end

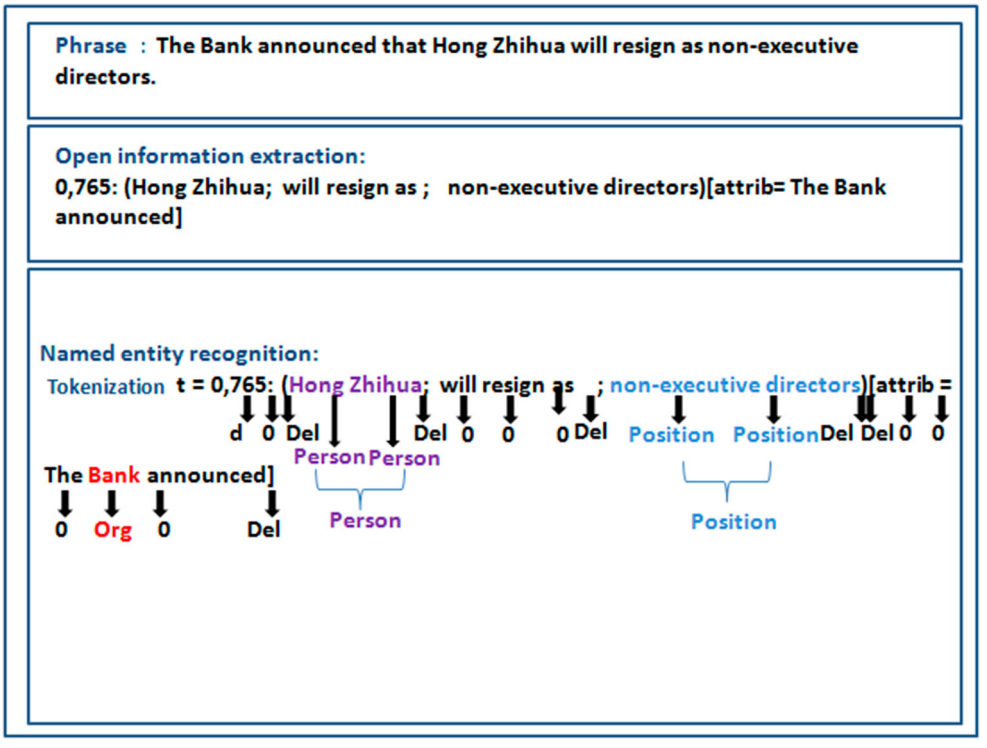

Figure 5. Example of recognition phase for the phrase 'The Bank announced that Hang Zhihua will resign as non-executive directors'.

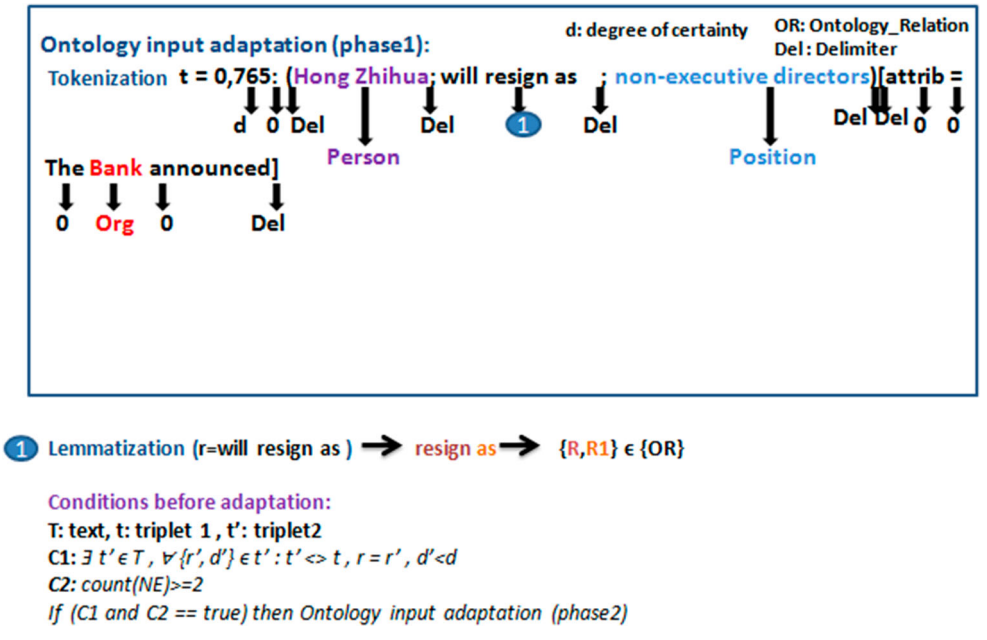

Figure 6. Ontology input adaptation for the second type of OIE system (phase1).

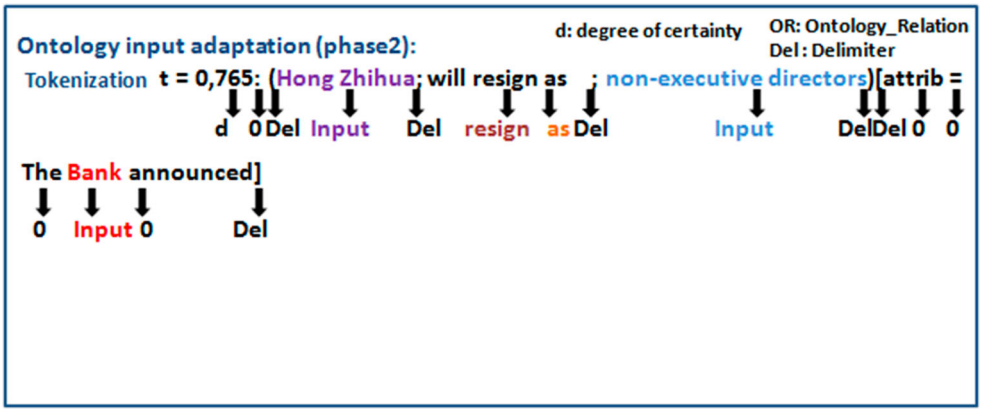

Figure 7. Ontology input adaptation for the second type of OIE system (phase2).

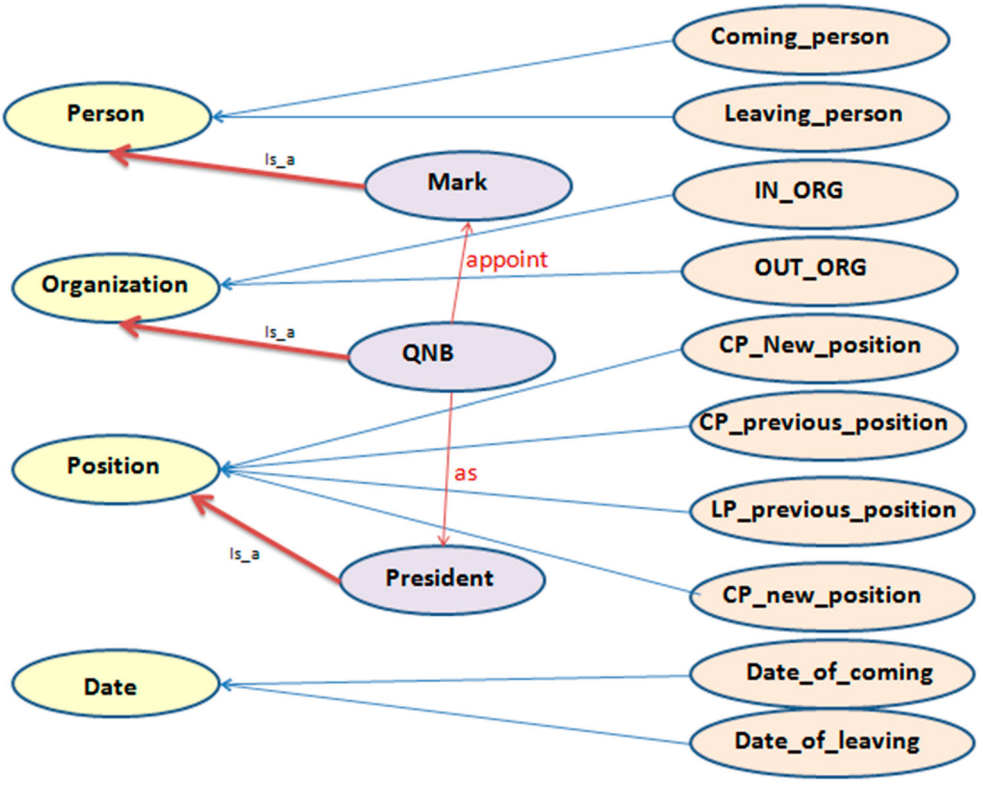

Figure 8. Example of the ontology after adaptation step for the sentence *'QNB appoints Mark as a president'*.

3.2.4. Step 4: reasoning

The reasoning is a stage after entering the instances and linking them by their specific relations. The reasoner is a software which infers logical consequences from a set of rules to affect for each instance its role (event). Ontology Web Language (OWL) is based on description logics, and it supports automated reasoning. Protégé OWL provides

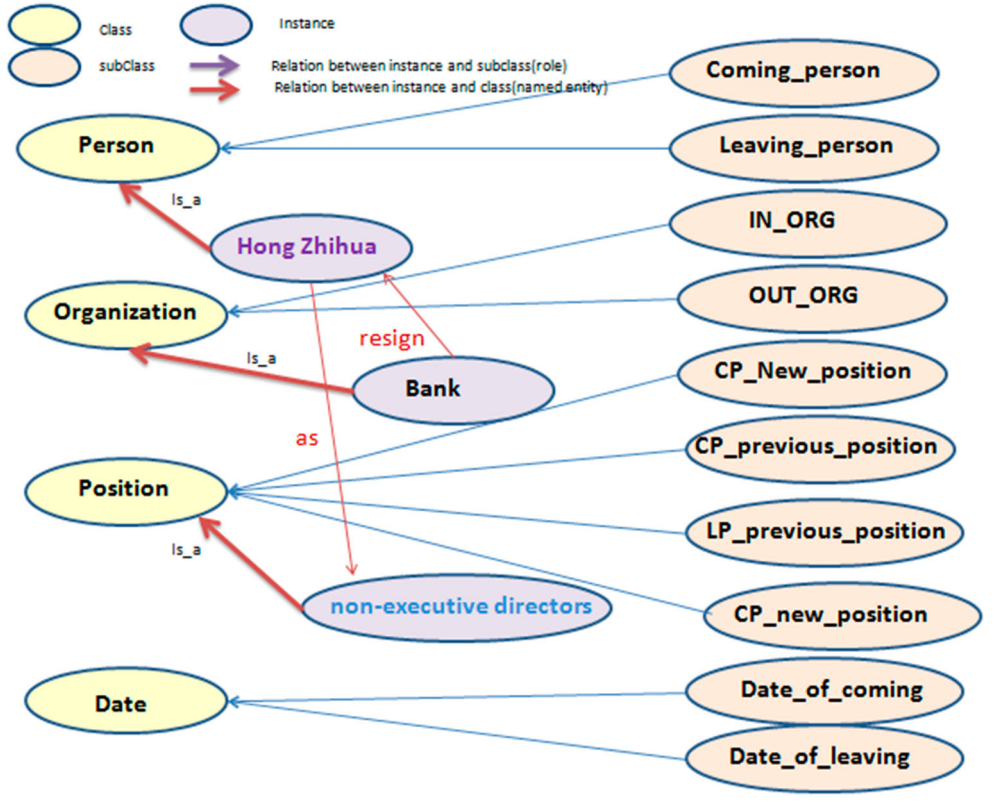

Figure 9. Example of the ontology after adaptation step for the sentence *'The Bank announced that Hang Zhihua will resign as non-executive directors'*.

direct access to reasoners such as HERMIT. The later can determine whether or not the ontology is consistent, identify subsumption relationships between classes, to say the least. The tokens that are recognized by named entities, i.e. QNB, Mark and president, will be entered as instances under the classes of the ontology and will be connected by their corresponding relationships, namely, appoint, as. This step is the matching phase who the process is explained by Algorithm 4.

As for example, in Figure 10, after reasoning, the entries (input1, input2, input3) are automatically linked by their roles (event). Hence, QNB has the role of IN_ORG, Mark has been assigned the role of Coming person and president has the role of CP_new_position as shown.

As a second example, as shown in Figure 11, the reasoner, through a set of rules, comes finally to the conclusion that 'Hang Zhihua' is a Leaving_person, 'non-executive directors' is a LP_previous_position and 'Bank' is an OUT_ORG.

4. Experimental results

The main objective of this work is to build a system for an automatic extraction of management change event from texts. We performed an evaluation to measure the application's quality for the event recognition on a set of a 'management change' corpora as described below.

Algorithm 4 Matching
Input
O : Ontology
tk : Tokens_input_ontology
R : Relation
Output
O2 : Ontology with instances linked by relations
Begin
if (O.Class == tk.NE) **then** *// if the named entity of the token matches a class of the ontology*
tk.addAsInstanceOf(O.Class) // the token will be added as instance under the class
tk.LinkInstancesBy(R)// the tokens will be linked by a relation R
end
O2 ← New_OntologyWith(tk, R)// a new ontology with instances tk linked by relation R
end

4.1. Dataset description

News about the management change event have been considered, as collected by RSS-Feed during the period from 15/08/2010 until 30/11/2011 from news providers in the scope of the Financial Watch Project (Jaoua et al., 2010). The collected data were analyzed and, manually, annotated by experts in order to build a multilingual platform for knowledge discovery, information extraction and automatic translation in the financial domain.

Management change is usually a strategic decision for corporations. Therefore, their occurrences are limited. Hence, the experiment was limited to 50 files for training and 40 files for testing. The experts have manually processed 919 sentences containing

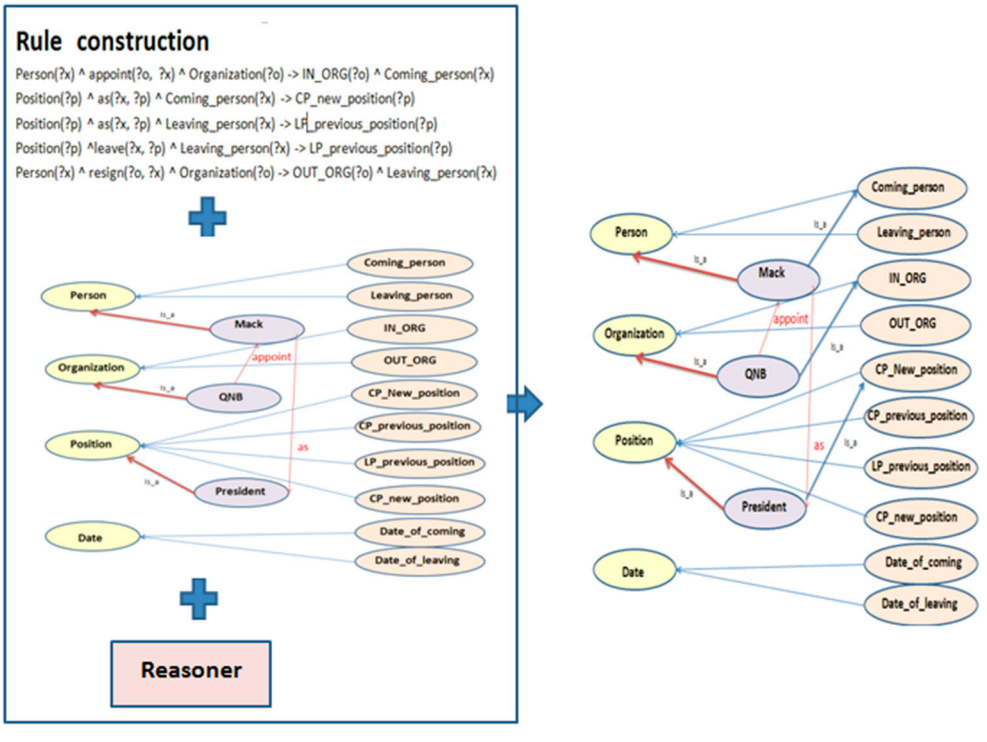

Figure 10. Example of the reasoning step and event extraction for the sentence *'QNB appoints Mark as a president'*.

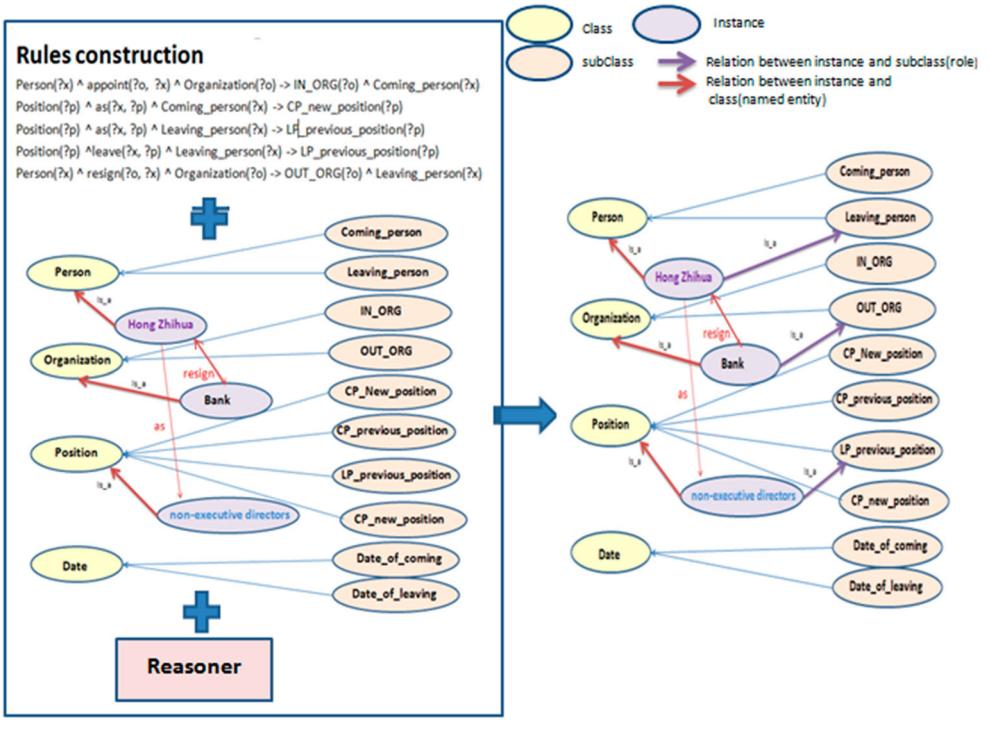

Figure 11. Example of the reasoning step and event extraction for the sentence *'The Bank announced that Hang Zhihua will resign as non-executive directors'*.

6995 words. In addition, the number of tags for the level 2 named entities is 338 as detailed in Table 2.

4.2. Test metrics

In our evaluation, we chose the metrics that will allow us to evaluate the results of our work and measure the performance of the proposed system. When the system returns an answer to a text and a class, two alternatives are available:

- The message belongs to the class.
- The message does not belong to the class.

This gives raise to four different possible cases:

(1) *True Positive (TP):* The system correctly finds the message as belonging to the class.
(2) *False positive (FP):* The system mistakenly finds the message as belonging to the class.
(3) *True negative (TN):* The system correctly finds the message as not belonging to the class.
(4) *False negative (FN):* The system mistakenly finds the message as not belonging to the class.

Precision is the measure of quality, recall is the measure of quantity and the F-measure is an aggregate of recall and precision. These three metrics were chosen for their frequent use in the field.

Table 2. Number of tags in management change dataset.

Level 2 NE	Description	#Tags
ORG	Organization of event	83
CP-NAME	Coming person name	91
Date-Coming	Coming date	18
CP-New-Position	Coming person new position	71
CP-Previous-Position	Coming person previous position	7
Date-Leaving	Leaving date	10
LP-Name	Leaving person name	36
LP-New-Position	Leaving person new position	22

Precision: Proportion of relevant solutions that are found. It asseses the ability of the system to provide all relevant solutions.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Proportion of solutions found that are relevant. It assesses the ability of the system to reject irrelevant solutions.

$$Recall = \frac{TP}{TP + FN}$$

F-measure: Measures the ability of the system to provide all relevant solutions and to deny others.

$$F{-}measure = \frac{2 \times P \times R}{R + P}$$

4.3. Results

The evaluation of our approach is divided into two parts, described in more detail in the following. At first, the results of a general evaluation of the recognition rate of all the events as given in Table 3. Next, the second part relates to the different types of events compared with two existing approaches.

The results in Table 3 for the last three columns, i.e. precision, recall and F-measure respectively, stand within 0–100%. For 30 files over 40, i.e. 75% of cases, we obtained more than 80% of precision, which is a very promizing approach performance indicator. However, in 4 cases the precision is equal to 0%. The reason is related to the OIE output, since we remarked that the triplets related to the targeted event were not detected. In average, the recall is 65.45% and only 6 cases the recall is less than 50%. This is due to the fact that some parts of the event were not extracted. In addition, for some cases, the NER output fails to identify the correct named entity of a text which implies a missing or wrong event role.

Moreover, Table 3 presents in detail the level 2 NEs number, for the different files in an ascending order. We remark that the precision is sensitive to the number of the NEs it contains. For instance, in the last three files there are more than 15 NEs and the system fails to recognize them correctly. However, the precision rate is more than 88% when the NEs number is less than 4.

We compared our work to a previous applications A-CRF and GLAEE (Elloumi, 2019). A-CRF is based on a correspondence between NEs and events. This correspondence is based

Table 3. Evaluation results for the 40 files.

ID file	NE1	NE2	NE3	NE4	NE5	NE6	NE7	NE8	Tot. NE	P (%)	R (%)	Fm (%)
F1	0	0	0	0	0	1	0	1	2	95.00	87.00	90.82
F2	1	0	0	0	1	0	0	0	2	100.00	50.00	66.67
F3	1	1	0	0	0	0	0	0	2	100.00	100.00	100.00
F4	1	1	0	1	0	0	0	0	3	88.00	79.00	83.26
F5	1	1	0	1	0	0	0	0	3	88.00	79.00	83.26
F6	1	1	0	1	0	0	0	0	3	88.00	79.00	83.26
F7	1	0	0	0	0	1	0	1	3	89.00	79.00	83.70
F8	1	1	0	1	0	0	0	0	3	90.00	79.00	84.14
F9	1	1	1	1	0	0	0	0	4	88.00	77.00	82.13
F10	1	1	0	1	1	0	0	0	4	88.00	77.00	82.13
F11	0	1	0	1	0	1	0	1	4	88.00	77.00	82.13
F12	1	1	0	1	1	0	0	0	4	88.00	77.00	82.13
F13	1	0	0	0	0	2	1	0	4	94.00	89.00	91.43
F14	1	1	0	1	1	0	0	0	4	94.00	94.00	94.00
F15	1	0	0	0	0	1	1	1	4	95.00	90.00	92.43
F16	1	1	1	1	0	0	0	0	4	100.00	33.00	49.62
F17	1	1	1	1	0	0	0	0	4	100.00	85.00	91.89
F18	1	1	1	1	0	0	0	0	4	100.00	88.00	93.62
F19	1	1	0	0	0	1	1	1	5	70.00	55.00	61.60
F20	1	0	0	0	0	1	1	2	5	74.00	60.00	66.27
F21	1	1	0	2	0	1	0	0	5	77.00	60.00	67.45
F22	1	1	1	1	0	1	0	0	5	83.00	66.00	73.53
F23	1	2	0	2	0	0	0	0	5	84.00	66.00	73.92
F24	2	1	0	1	1	0	0	0	5	87.00	66.00	75.06
F25	1	2	0	1	1	0	0	0	5	94.00	90.00	91.96
F26	2	1	0	1	0	1	0	0	5	100.00	57.00	72.61
F27	1	2	0	2	0	0	0	0	5	100.00	66.00	79.52
F28	2	1	0	1	1	0	0	0	5	100.00	75.00	85.71
F29	2	2	0	0	0	1	0	0	5	100.00	77.00	87.01
F30	1	0	0	0	0	2	1	1	5	100.00	80.00	88.89
F31	1	1	0	0	0	2	1	1	6	69.00	55.00	61.21
F32	2	1	0	2	1	0	0	0	6	81.00	55.00	65.51
F33	2	1	1	1	1	0	0	0	6	85.00	54.00	66.04
F34	1	2	0	1	1	1	0	0	6	88.00	100.00	93.62
F35	2	2	1	2	0	0	0	0	7	0.00	0.00	–
F36	2	2	1	1	1	0	0	0	7	76.00	55.00	63.82
F37	4	5	1	4	0	0	0	0	14	70.00	62.00	65.76
F38	4	2	1	5	3	0	0	0	15	0.00	0.00	–
F39	4	2	1	5	3	0	0	0	15	0.00	0.00	–
F40	2	5	2	3	3	2	0	2	19	0.00	0.00	–
Tot. NEs	**56**	**50**	**13**	**47**	**20**	**19**	**6**	**11**		**80.28**	**65.45**	**79.34**
										Av. P	Av. R	Av. F-m

Note: This table presents in detail the level 2 named entities number for the different files where : NE1: ORGANIZATION_OF_EVENT, NE2 : CP_NAME, NE3: DATE_COMING, NE4: CP_NEW_POSITION, NE5: CP_PREVIOUS_POSITION, NE6: LP_NAME, NE7: DATE_LEAVING, NE8: LP_NEW_POSITION, P:precision, R: recall and Fm:Fmeasure.

on a double generation of the classifier: A classifier for the first level's learning (PERSON, POSITION, ORG) and another classifier for the second level's learning (COMING PERSON, NEW POSITION, IN_ORG). GLAEE is based on a list of keywords and cue words to identify events. For the comparaison, we chose to compare our results against A-CRF and GLAEE by role as indicated in Table 4. Precision is considered to be a comparative metric between these two approaches.

Form Table 4, we underscore that our approach outperforms its competitors for the comingPerson and LeavingPerson event's roles. The presicion value for the DateOfComing is 33% is justified by OIE output which focus more on extacting verbs, and actors rather date related information. Nevertheless, the most important parts of a management change event were extracted.

Table 4. Comparative study.

Role	Our approach (%)	A-CRF (%)	GLAEE (%)
Coming_Person_Name	81.00	81.00	66.66
Date_of_Coming	33.00	69.00	88.88
CP_New_Position	76.00	81.00	84.90
CP_Previous_Position	76.00	81.00	84.90
Leaving_Person_Name	92.00	49.00	33.33
Date_of_Leaving	50.00	61.00	60.00
LP_New_Position	75.00	60.00	82.35

Even though quite encouraging, we can deduce that we arrived with an acceptable rate to extract from the texts a set of events. The A-CRF approach based on CRF analysis that requires human and manual intervention at the classifiers generation and which take a long time. The GLAEE approach involves a list of keywords and cue words that are manually acquired to identify events by an alignment between the pattern and the new text. Our approach can be considered as an automatic one except the part of the rule construction and the ontology which are useful for any event in the management change domain.

5. Conclusion

In this paper, we introduced an event extraction system from textual documents which is mainly domain dependent. The originality of our approach is to avoid as much as possible this dependency since we proposed to apply the Open Information Extraction approach for modelling any event type ontology representation.

The matching between the OIE output and the ontology requires adaptation and reasoning phases. Our approach is generalized to any type of text in the management change event in particular for the modal of the ontology and the OIE context.

The exprimental results confirms the succefulness of our approach since it was able to extract events with an accuracy rate comparable to A-CRF and GLAEE approaches.

We consider as future work, by merging between OIE and open domain event extraction approach (Elloumi, 2019). Based on the idea of the open domain, it can give us more effective results. In addition, we plan to use conjugated verbs instead of the lemmatized ones and to assign higher importance to the attribution/condition part to resolve the problem of two different roles in the same sentence and to identify the temporal order of statements.

Notes

1. https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/
2. https://protege.stanford.edu/
3. The OLLIE tool is available at this address: https://github.com/knowitall/ollie
4. https://www.ekino.com/articles/handson-de-quelques-taches-courantes-en-nlp

Acknowledgements

This work was made possible thanks to the Astra funding program Grant #2014-2020.4.01.16-032.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was made possible thanks to the ASTRA funding program Grant #2014-2020.4.01.16-032.

Notes on contributors

Sihem Sahnoun received the master's degree fromthe Faculty of Sciences of Tunis, Tunisia in 2019. She is currently a phd student in Enet'Com University. She has focused on the domain of IA and contributed in severalaspects related to information extraction and deep learning.

Samir Elloumi received the Habilitation to Lead Researches in Computer Sciences fromthe Faculty of Science of Tunis, Tunisia, in April 2019 and the PhD., in June 2002. He has been an assistant professor with the University ofJeddah (KSA), since September 2019,and the University of Tunis ElManar, since September 2003. His research interests mainly include conceptualmethods for data structuring, information extraction and multimedia learning.

Sadok Ben Yahia received the Habilitation to Lead Researches in Computer Sciences fromthe University of Montpellier, in April 2009. He has been a Professor with the Tallinn University of Technology (TalTech), since January 2019,and the University of Tunis ElManar, since September 2001. His research interests mainly include combinatorialaspects in big data and theirapplications to different fields,e.g., data mining,combinatorialanalytics (e.g., maximumclique problem and minimal transversals),and smart cities (e.g., information aggregation & dissemination and traffic prediction).

ORCID

Samir Elloumi http://orcid.org/0000-0002-1822-5334
Sadok Ben Yahia http://orcid.org/0000-0001-8939-8948

References

Abuleil, S. (2006). Hybrid system for extracting and classifying arabic proper names. *Conference on artificial intelligence, knowledge engineering and data bases* (pp. 205–210). Madrid-Spain.

Agichtein, E., & Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pp. 85–94.

Angeli, G., Tibshirani, J., Wu, J., & Manning, C. D. (2014). Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar. pp. 1556–1567.

Benajiba, Y., Diab, M., & Rosso P, P. (2009). Using language independent and language specific features to enhance arabic named entity recognition. *International Arab Journal of Information Technology*, *6*(5), 464–473.

Berland, M., & Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics, College Park, Maryland, USA*. pp. 57–64.

Brin, S. (1998). *Extracting patterns and relations from the world wide web*.

Chiu, J. P. C., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4, pp.357–370.

Durme, B. V., & Schubert, L. K. (2008). Open knowledge extraction through compositional language processing. In *Proceedings of the Conference on Semantics in Text Processing*, Venice, Italy, pp. 239–254.

Elloumi, S. (2019). An adaptive model for sequential labeling systems. *Multimedia Tools and Applications*, *78*, 22183–22197. https://doi.org/10.1007/s11042-019-7558-8, ISSN:1573-7721

Elloumi, S., Jaoua, A., Ferjani, F., Semmar, N., Besancon, R., Al-Jaam, J., & Hammami, H. (2013). General learning approach for event extraction: Case of management change event. *Journal of Information Science*, *39*(2), 211–224. https://doi.org/10.1177/0165551512464140.

Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. *Proceedings of the conference on empirical methods in natural language processing*, Edinburgh, Scotland, UK (pp. 1535–1545).

Grishman, R., & Sundheim, B. (1996). Message understanding conference-6: A brief history. Proceedings of the 16th conference on computational linguistics, Copenhagen, Denmark (pp. 466–471).

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th conference on computational linguistics COLING '92*, Copenhagen, Denmark (pp. 539–545).

Jaoua, A., Jaam, J., Hammami, H., Ferjani, F., Laban, F., Semmar, N., Essafi, H., & Elloumi, S. (2010). Financial events detection by conceptual news categorization. *Proceedings of the international conference on intelligent systems design and applications (ISDA'2010)* (pp. 1101–1106). Cairo, Egypt.

Kodelja, D., Besancon, R., & Ferret, O. (2017). *Représentations et modèles en extraction d'événements supervisée* (pp. 1–7). CEA, LIST.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, Lake Tahoe, Nevada, USA, (pp. 3111–3119).

Nguyen, T. H., & Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Berlin, Germany, pp. 39–48.

Niu, F. (2012). Elementary: large-scale knowledge-base construction via machine learning and statistical. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 1–23.

Sarawagi, S., & Cohen, W. (2004). Semi-Markov conditional random fields for information extraction. *NIPS'04: Proceedings of the 17th International Conference on Neural Information Processing Systems*, Vancouver, USA (pp. 1185–1192).

Satoshi, S., Kiyoshi, S., & Chikashi, N. (2002). Extended named entity hierarchy. ', Las Palmas, Canary Islands (pp. 1818–1824).

Shaalan, K., & Raza, H. (2009). NERA: Named entity recognition for arabic. *Journal of the American Society for Information Science and Technology*, *60*, 1652–1663. https://doi.org/10.1002/asi.21090.

Talha, M., Boulaknadel, S., & Aboutajdine, D. (2014). Système de Reconnaissance des Entités Nommées Amazighes. *actes de TALN*. (pp. 517–524).

Vo, D. T., & Bagheri, E. (2017). Open information extraction. Encyclopedia with semantic computing and Robotic intelligence 1, no. 01 (2017): 1630003.