Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ONTOLOGY-BASED ENHANCEMENT OF RULE LEARNING FOR INFORMATION EXTRACTION

[1]FATINE JEBBOR, [2]LAILA BENHLIMA
AMIPS Team, Computer Science Department of School Mohammadia of Engineers, Mohammed V University in Rabat, Morocco
E-mail: [1]fatinejebbor@research.emi.ac.ma, [2]benhlima@emi.ac.ma

ABSTRACT

Several data sources like social networks and blogs are providing increasing amounts of unstructured data in natural language. This data contains useful information that must be identified automatically, quickly and with high precision. Therefore, different Information Extraction (IE) approaches were proposed like Rule-based and Statistical ones. The majority of rule learning IE systems present a recurring problem caused by the generation of a set of irrelevant and unnecessary rules, which affects the quality of the extraction results. Hence, we propose in this paper a novel and generic approach to increase the performance of these extractors in order to avoid missing important information or providing erroneous one. It consists in enhancing the rule generalization through using a domain ontology designed to make the systems able to generate only the most likely useful rules. To prove its efficiency, our solution is applied to $(LP)^2$ system and empirically tested on a corpus for seminar announcements. According to our results, the system's enhanced version reaches a high accuracy with respect to $(LP)^2$ and other extractors, which means that the information it extracted is of a better quality.

Keywords: *Information Extraction, Rule-Based Learning, Natural Language Processing, Ontology*

1. INTRODUCTION

Nowadays, unstructured data, especially the textual type, know rapid expansion that is in continuous growth. This concerns for example social networks, digital libraries, blogs and other giant sources that provide increasing volumes of textual data. These texts often contain interesting and valuable information that can be very helpful to the decision-making process. However, searching such information in a large collection of texts is a hard task that requires a lot of time and effort. In other words, finding automatically and quickly information of high precision is a real challenge [1]. This explains the interest of researchers in Information Extraction discipline that offers solutions allowing the automatic identification of highly useful information from masses of unstructured or semi-structured data, to deal with issues like Question-Answering [2]. These solutions are called Information Extractors or Annotators. Indeed, the terms extraction and annotation are closely related because the extraction process consists in annotating the text by inserting tags to delimit the information one is looking for. For instance, if the user needs to know the birthplace and the nationality of an author, the corresponding entities will be tagged in the text as shown in the following example: *Elizabeth Strout was born in <Birthplace> Portland </Birthplace>. She is an <Nationality> American </Nationality> novelist and author*.

Information Extraction (IE) task is non-trivial because of the richness and the complexity of natural language [3]. Actually, the same fact can be expressed in different ways, and sometimes, relevant information is implicit. For this reason, IE involves in general the use of some Natural Language Processing (NLP) techniques for preprocessing the free text, like Tokenization [4] that consists in dividing the latter into basic units and Part-of-Speech (POS) Tagging [5].

IE researchers proposed two main approaches for building Information Extractors: Rule-based and Statistical. The first category consists in using a set of extraction rules constructed manually or learned automatically from a tagged corpus. In the second approach, the IE task is viewed as a sequence-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

labeling problem [6, 7] where the text is considered as a sequence of observations (words) and the system must assign a tag to each one. Both categories have strengths. On one hand, Statistical IE systems have shown great robustness to noise in unstructured data. Besides, they are more convenient in open domains like opinion extraction from social media. On the other hand, Rule-based IE systems are easier to comprehend and maintain [8]. Indeed, using rules facilitates both error analysis and the extension of these extractors. In addition, rules are a natural way of modeling the human perception to solve problems. In a number of works, Rule-based IE approaches have outperformed some Statistical ones for tasks like temporal expression detection. For instance, the temporal tagger HeidelTime [9] scored the best results with respect to NavyTime [10] which is based on a Maximum Entropy classifier.

As part of Natural Computation research, which is focused on modelling and explaining aspects of human intelligence, our study attempts to understand and process the human natural language in order to extract or predict important information from text. Besides, it is based on a conclusive research because it involves the design and implementation of a final and conclusive solution to the problem of lack of accuracy in existing systems. More specifically, we tackle in this paper, a weakness which is common to rule learning IE systems, for the task of Named Entity Recognition (NER). In other words, we aim to increase their performance/accuracy by making them able to avoid the generation of a wide range of undesirable extraction rules. To illustrate its efficiency, our solution is applied to (LP)2, a rule-learning IE system that is enhanced/upgraded to a new version called E-(LP)2. We will prove through this paper, that our contribution can improve the quality of the information extracted and therefore provide a useful Extractor that can be used in different applications like Question Answering and Sentiment Analysis.

The rest of our paper is organized as follow. We start by briefly outlining the principle of Rule-based Information Extraction approaches. Then, we present and discuss the related works. The next section is dedicated to the problem setting. Afterwards, our approach for enhancing the rule Generalization, is detailed. Next, the experiments conducted to test our solution are reported and discussed. Finally, we end with a conclusion and future work.

2. RULE-BASED INFORMATION EXTRACTION

Information Extraction is a complex process that consists in identifying relevant, precise and useful information from textual data sources. It goes beyond Information Retrieval which is a traditional method relying on using keyword-based search like in search engines that merely return a set of documents to the user and delegate to him the task of seeking the desired information [11]. This is why there is a growing demand on Information Extractors. Indeed, they are becoming the basis for a great variety of enterprise applications [12]. These systems can be categorized along two dimensions [13]. The first one contrasts Hand-coded and Learning-based approaches. Hand-coded ones are based on the manual generation of rules or regular expressions, whereas Learning-based extractors perform the training of Machine Learning models on manually tagged corpus. The second dimension distinguishes between the two basic types of IE techniques: Rule-based and Statistical, as we mentioned in Section I.

The majority of early IE systems were based on hand-coded rules like in Proteus [14]. They consist in general of two components: a collection of rules and a set of policies to monitor their firing. The extraction rules are composed of specific patterns developed by human experts using dedicated languages like JAPE [15]. This type of extractors can achieve good performance when they are applied to a specific target domain. However, designing good extraction rules is time-consuming and labor intensive [16]. Therefore, researchers move towards developing systems that automatically learn these rules by applying Machine Learning techniques on pre-tagged corpus. The general process of the majority of Rule-based learning IE systems is given in Figure 1.

The process starts by constructing a collection of preliminary or Initial Rules (IR) from instances contained in the training corpus. In general, a rule is defined by the relation Condition⇒Action. The Condition contains information about the attributes of text tokens (like their POS tags); these are the requirements that must be fulfilled to make the rule applicable. Once the Condition matches a sequence of words in the text, this implies the execution of the Action that consists in inserting tags in the text to delimit the desired information.

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

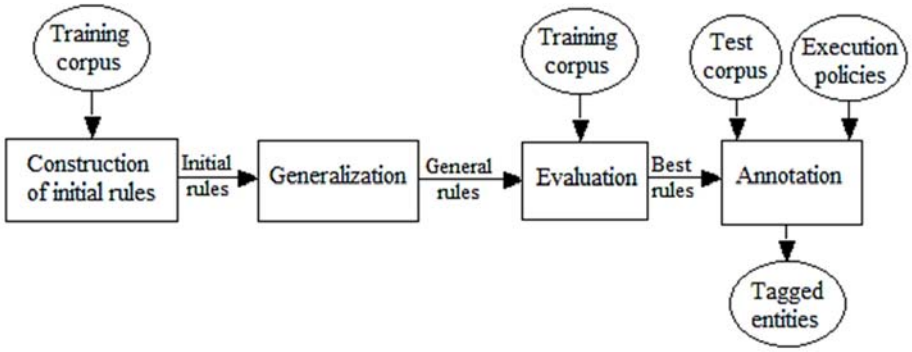

Figure 1: The General Process of Rule-Learning Information Extraction Systems

Table 1: Example of an IR Generated to Identify Names of Persons Giving a Speech in a Conference.

Token index	Condition				Action (Tag)
	String	Lemma	POS tag	Case*	
1	the	the	DT	Low	
2	26th	26th	JJ	Low	
3	March	March	NP	Up	Insert
4	Mary	Mary	NP	Up	<Speaker>
5	Lutz	Lutz	NP	Up	after
6	presents	present	VVZ	Low	Token 3
7	a	a	DT	Low	
8	talk	talk	NN	Low	

* Case indicates if the word begins with upper or lower case letter

This can be illustrated by the following example.
Example 1:
"On the 26th March, <Speaker> Mary Lutz </Speaker> presents a talk about women in science."

In this sentence, "*Speaker*" is called target concept, and "*Mary Lutz*" which is an instance of *Speaker*, is called positive example. We name Initial Snippet (IS) the text extract consisting of the positive example and a number (n) of words in its neighborhood. For example, for n = 3, the IS is: "*the 26th March, <Speaker> Mary Lutz </Speaker> presents a talk"*. An Initial Rule corresponding to the latter is described in Table 1.

After the generation of the IRs, Rule-based learning systems perform their Generalization to produce a tree gathering a set of General Rules (GRs) able to extract other instances of the target entity. This process will be detailed and illustrated later. Afterwards, every GR is evaluated on the training corpus to keep only the relevant ones which will be applied to the test corpus to perform the annotation. Finally, the accuracy is calculated.

Having outlined the principle of Rule-based IE learning systems, we move to present and discuss the related works.

3. RELATED WORKS

Rule-based systems have a long history and the majority of them are interesting since they present different and rich strategies to learn rules from labeled data. Among the most popular works, we find TIPSI [17], (LP)2 [18], SEE [19], RAPIER [20], BWI [21], etc. To the best of our knowledge, there are no more recent systems. However, rule learning technology has many strengths and is widely used in in the commercial world [22]. RAPIER lays on Relational Learning [23] and uses techniques from Inductive Logic Programming (ILP) [20]. It starts by generating specific rules from documents to fill in the slots of a predefined template. In (LP)2, the rule induction process begins by considering a positive example in the training corpus and producing the corresponding Initial Rule. However, this extractor dismisses some important words constituting the IE context because the extraction window is not correctly defined to form the Initial Snippet, which leads to information loss. Indeed, this window covers n tokens before and after

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the opening tag in the training corpus [24] and ignores those located after the closing tag. However, TIPSI for example, whose role is to assign metadata to semi-structured documents via IE, selects dynamically the appropriate window size for each target entity [17]. We consider this technique as an advantage of the second system, because it allows to choose the window size that gives good results by evaluating the performance of all possible cases. Nevertheless, this is a time consuming process; the learning is performed for all the IRs constructed according to many window sizes [17].

After the IR generation, rule learning systems try to make it more general in order to cover additional instances. In TIPSI for example, the learning process relies on Similarity-based Rule Learning (SRL) [17] and consists of iterations; in each one, pairs of the most similar IRs are selected for Generalization. Next, the obtained annotation rules undergo evaluation before application to unlabeled documents. Regarding (LP)2, the IR is generalized in a Bottom-up way and the k best generalizations are selected [25]. Then, retained best GRs are applied to extract the desired information. In RAPIER, rule induction is also performed based on a Bottom-up learning strategy and the annotation rules are in the form of patterns that extract fillers from text documents.

An important difference between $(LP)^2$ and TIPSI is that the latter generates at the beginning all the IRs even those that are similar to existing ones or those covered by others. Then, it generalizes them by pairs. Consequently, the Generalization process takes more time compared to (LP)2 that, once the first IR is generalized, it removes from the training corpus all positive instances covered by it, because there is no need to generate new IRs for them. Regarding RAPIER, we notice that the real difference between it and these two systems lies in rule Generalization step where it uses a rule compression algorithm which can sometimes generate some strange rules having no sense [24].

SEE is relatively a new system that gives competitive results. It deals with the task of record extraction from text documents. The process consists of detecting entity mentions and converting them into tuples to populate the target relations in a Bottom-up way [19]. For this purpose, Discriminative Structured Learning (DSL) is applied to correctly predict a structured entity (populated relational Schema) for each input text document. We notice that many techniques depending on the corpus were used in this system, such as features, some hand-crafted rules and dictionaries [19]. Therefore, if one wants to apply this approach to a different corpus, a big part of the work will have to be redone.

Regarding the works adopting a statistical IE approach, Word2vec [26] and FastText [27] are among the popular systems. They are based on Word Embedding which is a natural language modelling technique used to map each word from a vocabulary to its corresponding vector of real numbers in a predefined vector space.

Word2vec treats the input text as a bag of individual word vectors that are learned from large text corpus. It has several useful applications such as Named Entity Recognition [28]. However, its main weakness is its inability to provide representations for words that are not contained in the training corpus even if the latter is large and including more vocabulary. Besides, Word2vec considers each word as a single entity and ignores its morphological structure [29].

FastText is a simple neural approach proposed by Facebook for text representation and classification. Its principle is similar to that of word2vec, but instead of considering individual words, FatsText breaks the latter into several n-grams. For example, the tri-grams corresponding to the word "*Dream*" are *<Dr, Dre, rea, eam, am>*, where "<" and ">" are boundary symbols. The embedding vector of each word is the sum of all its n-grams vectors. The advantage of this system over Word2vec is that embeddings for words that occur rarely in the training corpus are of better quality [29]. However, its limitation lies in its weak sentence representation [30]. More specifically, the calculated mean of word vectors is not weighted, which makes the representation of words such as "*a*" or "*the*" contribute equally to the representations of important words. In addition, the performance of both FastText and Word2vec decreases in the case of reduced corpus size [31].

To sum up, each of the systems we have cited has advantages and weaknesses that can be serious depending on the extraction task. The main weakness that we are interested in, in rule learning IE systems, is related to rule Generalization. In the following section, this problem will be detailed.

4. PROBLEM SETTING

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

After having studied several state-of-the-art Rule-based IE algorithms for NER, we noticed that successful systems like (LP)2 and TIPSI, can give even better results if we improve them by addressing their the weakness they have in common. More specifically, during the Generalization, these extractors generate from the Initial Rules a large number of general ones. For instance, an extract from the tree gathering the General Rules corresponding to the IR described in Table 1, is depicted in Figure 2 at the end of the paper. Each rectangle represents a GR. According to the rule *[<Speaker> NP NP presents*] for example, if the system finds in the text the word "*presents*" preceded by two successive words whose POS tags are NP (Proper Noun), then it will insert the tag *<Speaker>* just before the first word. The construction process of this tree is explained in [18].

After having generated the GRs, the system processes each one by calculating some metrics to decide whether to retain it or not for executing the annotation. However, among these rules, several ones are irrelevant or useless and mustn't be produced. For instance, those located in the first column of the first three levels in the tree above, are all irrelevant because they don't specify any condition indicating that the text is talking about a Speaker, such as the presence of a person's name or the action of giving a speech in a conference. Unfortunately, IE systems generate such rules and after spending a considerable time in their processing, they find later that several among them mustn't be retained because they have a low accuracy.

In the following section, we address this limitation and propose an ontology-based generic solution to make rule-learning systems able to not generate irrelevant or useless rules during the Generalization.

5. ENHANCEMENT OF RULE GENERALIZATION BASED ON A DOMAIN ONTOLOGY

The goal of our contribution is to enhance Information Extraction and better satisfy the user need. More specifically, we aim to make Rule-based systems generate only the most likely useful General Rules (GRs). Actually, each generated IR can contain general tokens which aren't necessarily related to the target concept. These tokens lead to the production of irrelevant and useless GRs. Hence, the general idea of the proposed approach is to assess the relevance, based on a domain ontology, of the tokens constituting each Initial Snippet from which an IR will be generated, and remove the irrelevant ones from it to ensure a better generalization of the IR. Therefore, a considerable set of undesirable rules won't be produced.

This section is organized as follow. We start by presenting the domain ontology we designed to execute our solution. Then, the process of the latter is explained. Finally, we outline through a use case the asset of the proposed approach and prove its efficiency.

5.1 OntoSA Ontology

5.1.1 Background

The term ontology refers to a formal representation of a shared understanding about a given knowledge set. It reduces terminological and conceptual confusion by providing a unifying conceptual framework for the different assumptions [32]. Among the well-known types of ontologies, are those dedicated to model a specific domain knowledge, called "domain ontologies".

In general, an ontology gives a description of a set of concepts, their definitions and their relationships between each other. Its encoding is performed via formal languages like OWL which is widely used in many application domains such as defense, biology, etc [33]. There are several examples of well-known ontologies, like Wordnet, DOLCE and SNOMED [34]. In order to model an ontology, many editors are used such as Protégé [35] which was developed by Stanford University.

5.1.2 OntoSA Description

To improve the rule Generalization, we designed OntoSA which is dedicated to the domain of seminars. An extract from this ontology, focused on the concept Speaker, is presented in Figure 3. In the following, we give a brief conceptual description of this extract in terms of classes and relationships.

The extract contains ten fundamental classes: Seminar, Speaker, Person, Diploma, Profession, Biography, ContractedTitle, Audience, Organizer and Entity. These classes are linked to each other by relationships (Object Properties), and some of them has one or many subclasses. We present in what follow a brief description of each class.

Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

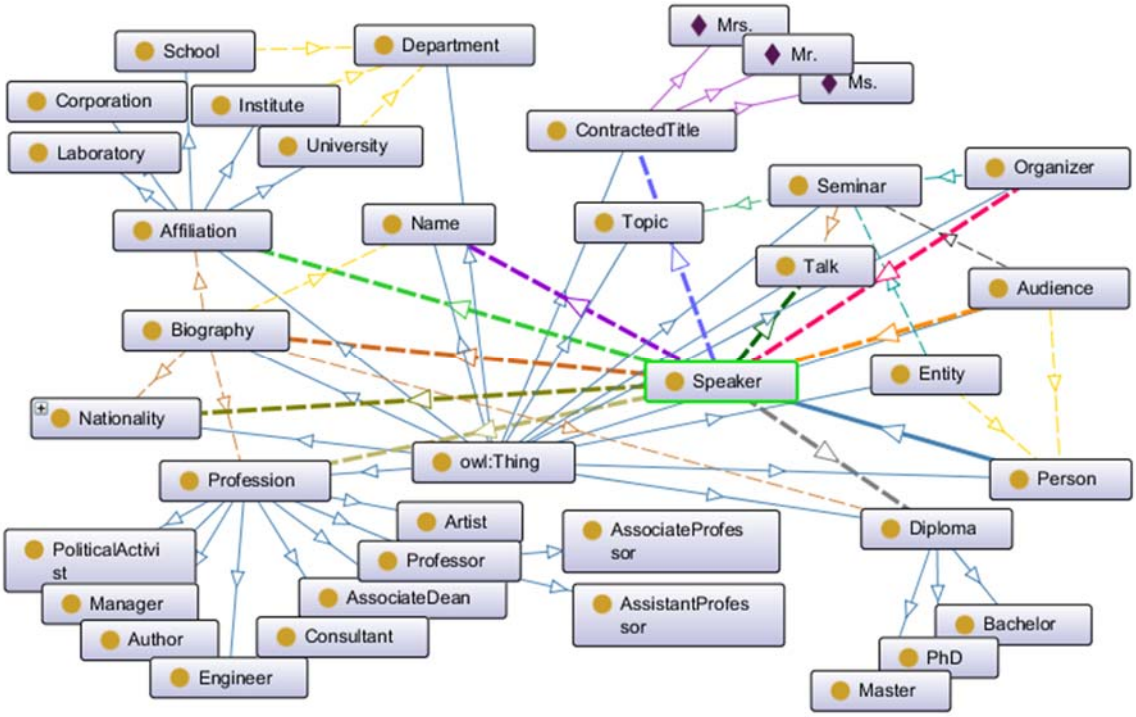

Figure 3: Extract From OntoSA Ontology Focused on the Speaker Concept

- **Seminar**: is a meeting between one or more experts and a group of people to discuss something. It has exactly one topic (**Topic**) and involves several speeches (**Talk**).
- **Speaker**: is every person (**Person**) who intervenes to give at least one speech during a Seminar. Since several terms expressing the Speaker concept can be used in the text, we added to the ontology some synonyms like Lecturer.
  Every Speaker has a set of characteristics. For instance, he has a biography (**Biography**) indicating some information like its **Name**, **Nationality**, **Diploma**, **Profession** and **Affiliation**. More specifically, a speaker has exactly one Name, he received one or more Diploma and he practices at least one Profession. Besides, he is member of at least one Affiliation which can be a school (**School**), a laboratory (**Laboratory**), a society (**Corporation**), etc. In addition, he is called using one contracted title (ContractedTitle) such as Mr., Mrs, and Ms..
- **Audience**: is a group of people who attend at least one given Seminar. It must contain at least five people.
- **Organizer**: is the responsible for organizing one or more seminars. An organizer can be an Entity or a Person.
- **Entity**: is a group of individuals (at least 3 people) with a collective goal. It can be an association, a company, an organization, etc.

Having outlined OntoSA, we move to present our solution.

5.2 Process of The Proposed Approach

Our solution consists in enabling Rule-based systems to evaluate the relatedness between the target concept and the tokens of each Initial Snippet, based on OntoSA ontology, in order to remove from it some irrelevant or unnecessary words whose presence in the IR leads to the generation of undesirable rules. The process of the proposed enhancement is defined by the following steps, as depicted in Figure 4:

5.2.1 Construction of Set_i

In this first step, the system takes as input the Initial Snippet corresponding to the positive example and constitutes two sets (Set_1 and Set_2) representing the context of the concept.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

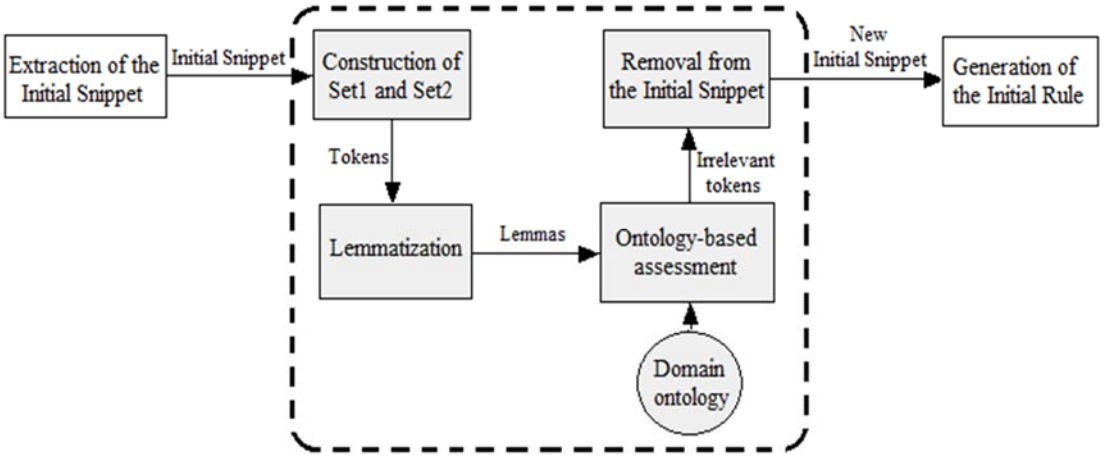

Figure 4: General Process of our approach for the Generalization Enhancement

- Set_1: consists of the IS tokens located before the opening tag.
- Set_2: contains the IS tokens located after the closing tag.

The elements of these sets undergo the next step.

5.2.2 Lemmatization

Each word (or token) constituting Set1 and Set2 is lemmatized in order to get its base or dictionary form known as Lemma.

5.2.3 Ontology-based Assessment

At this stage, IE systems evaluate the relatedness between the target concept and each word constituting the sets Set_i. This consists in checking whether these words are closely related to the concept or not. Indeed, a strong relatedness between a given word and the concept means that it's highly probable that an instance of the latter be cited in the proximity of this word. For this purpose, the system checks whether the lemmatization results of Set_1 and Set_2 tokens identify elements of the proposed domain ontology. In other words, it searches every lemma among the classes, subclasses, labels, individuals and object properties defined in OntoSA. Indeed, if a lemma is identified for example as an OntoSA class, the relatedness between the concerned word and the target concept will be considered strong. Afterwards, the system executes the appropriate action depending to two cases. If there is at least one word from Set_1 which is strongly related to the concept, then all of Set_1 words will be retained to constitute the IR and participate to its Generalization. Otherwise, all the words of this set will be removed from the IS. The same process is applied to Set_2 words.

We would underline that the elimination of the irrelevant tokens does not concern the instance of the concept in the Initial Extract; It applies only to the tokens of Set_i.

Our solution can be represented in terms of instructions by Algorithm 1. For simplification reasons, we assume that the IS contains $n=3$ tokens before the opening tag and after the closing one. This algorithm remains valid for other values of the parameter n.

Algorithm 1: Initial Snippet pruning for enhancing the Generalization

```
Function NewInitialSnippet(arg1 IS, arg2
                           Concept, Ontology)
Begin
  InitialSnippet IS
  TokenizedIS←NLPprocessor.Tokenizer(IS)
  NumberTokens←NLPprocessor.CountTokens(
                                 ToknizedIS)
  Set1←{NLPprocessor.getToken(IdToken=i)}i
                                 from 1 to 3
  Set2←{NLPprocessor.getToken(IdToken=i)}i
       from NumberTokens-2 to NumberTokens
For j from 1 to 2
  For k from 1 to 3
    Lemk← NLPprocessor.Lemmmatizer(
                                 Setj.Tokenk)
    Boolean Rk← AssessRelatedness(Lemk,
                           Concept, Ontology)
  EndFor
  RSet← {Rk}
  If (∃ Rk ∈ RSet) / Rk=true Then
    Retain(Setj.Tokens, IS)
  Else
    Remove(Setj.Tokens, IS)
  EndIf
  j←j+1
EndFor
Return IR
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
28: End

29: AssessRelatedness(arg1 Lem, arg2 Concept,
30:                                 Ontology)
31: Begin
32:    Boolean S← Search(Lem, Ontology)
33:    If S=true then
34:        return true
35:    Else return false
36: End
```

NewInitialRule() is the main function that performs the IS pruning. The function *AssessRelatedness()* is called in order to check if the lemma identifies an element of OntoSA.

5.3 Use Case and Asset of The Proposed Solution

To clarify the process of our approach, reconsider Example 1 (*"On the 26th March, <Speaker> Mary George Lutz </Speaker> presents a talk about women in science"*). We remind that for *n=3* for example, the IS is: *"the 26th March, <Speaker> Mary Lutz </Speaker> presents a talk"*. The IS tokens constituting the sets Set_1 and Set_2, are presented and accompanied by their lemmas in Table 2.

Table 2: Constitution of Set1 and Set2 Corresponding to Example 1 and Token Lemmatization.

	Set_1			Set_2		
Tokens	the	26th	March	presents	a	talk
Lemmas	the	26th	March	present	a	talk

None of Set_1 tokens are identified in OntoSA, so they will all be removed from the IS. Regarding Set_2, the lemmas "*present*" and "*talk*" are an object property and a class of the ontology, respectively. Hence, all the terms of this set will be retained. Therefore, the resulting new IR is described in Table 3.

Table 3: The New IR Corresponding to Example 1, Generated After the Generalization Improvement.

Token index	Condition				Action (Tag)
	String	Lemma	POS tag	Case	
4	Mary	Mary	NP	Up	Insert
5	Lutz	Lutz	NP	Up	<Speaker>
6	presents	present	VVZ	Low	before
7	a	a	DT	Low	Token 4
8	talk	talk	NN	Low	

In the next step, this rule undergoes the Generalization process leading to the production of a new tree which is presented in Figure 5.

The asset of our solution can be proved using the new Generalization tree depicted in Figure 5. Indeed, it contains 82 GRs in the first four levels, while the original one (Figure 2) contained 172. Hence, our solution allowed to eliminate the production of a set of 90 irrelevant GRs, i.e. more than the half of the original tree rules. These GRs are those colored in Figure 6.

In order to prove their irrelevance, we categorized the colored rules along the following five categories, as shown in Figure 6, based on an intuitive analysis.

- Category1: These rules don't specify any condition that may indicate a Speaker, such as the presence of a person's name or the action of giving a speech in a conference.
- Category2: These GRs require a specific proper noun (*Mary* and *Lutz*) among the conditions. It may refer to a person name, but not necessarily a Speaker, especially if the text also talks about the seminar organizers or people to contact for more details.
- Category3: Here, a word whose POS tag is NP, is considered as Speaker although the conditions on the words that precede it don't necessarily imply it. These rules can cause several annotation errors because NP can refer to other named entities like places, cities, etc.
- Category4: These rules are similar to those of Category3, except that they require two proper nouns instead of one. They may refer to a person name, but not necessarily a Speaker.
- Category5: These GRs are unnecessary because the instances they extract are already covered

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

by the rule [<*Speaker*> *NP NP presents*] (rectangle in bold) which is more general and relevant.

In this section, we present the application of our approach to a rule learning system, and its validation. We demonstrate that the proposed

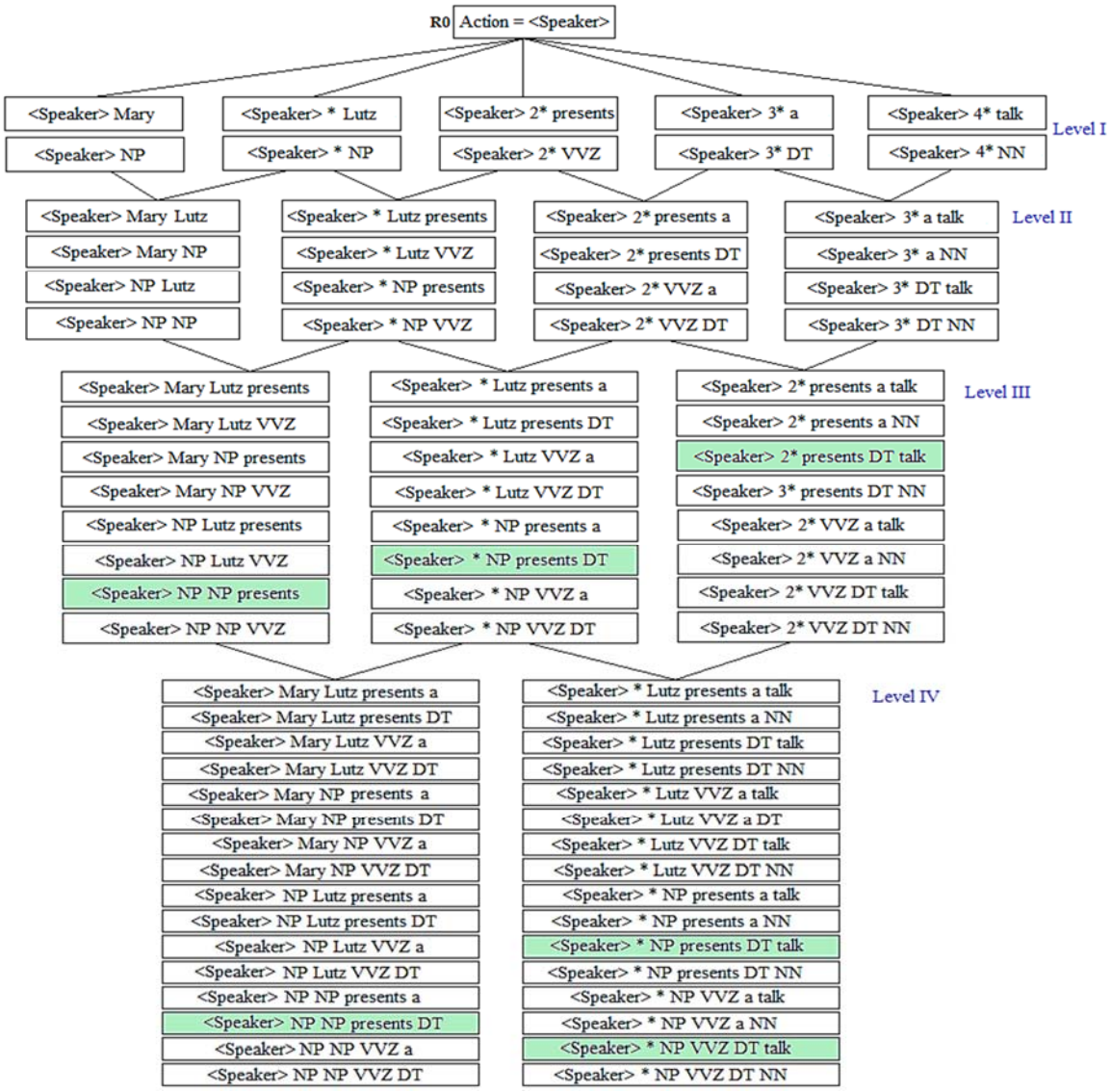

Figure 5: The New Generalization Tree Generated After Executing the Proposed Approach

This analysis shows that our solution eliminated the generation of several categories of undesirable rules, specifically those containing the tokens removed from the IS, and kept different GRs that perform the extraction efficiently and with fewer errors, like those colored in green in Figure 5. Furthermore, we present in the next section the test results which prove that our contribution increased the quality of the extracted information.

6. EXPERIMENTS AND VALIDATION

improvement can boost the system's accuracy. We begin by justifying the choice of the system. Then, the corpus is presented. Next, we describe the experimental methodology adopted to test the system's enhanced version. Afterwards, we report the obtained results in terms of classical measures and compare them with those of other IE systems. Then, these results are discussed. Before concluding, we briefly outline the validation of our solution.

6.1 Choice of The System

Among the systems we presented in Section III, we chose $(LP)^2$ to undergo our

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

experiments. This choice is motivated by the extraction results it obtained with respect to the other Rule-based systems. Indeed, the latter were tested on the CMU (Carnegie Mellon University) Seminar Announcements corpus and achieved the accuracies shown in Table 4.

Table 4: The Extraction Accuracy Obtained by (LP)², RAPIER, TIPSI and SEE on the CMU Seminar Announcements Corpus.

	Macro-Averaged F1 (%)
(LP)²	86.0
RAPIER	77.3
TIPSI	85.8
SEE	93.7

The best accuracy was obtained by SEE. However, we are not interested in the latter because it uses several tools built manually as explained in Section III. Since (LP)² obtained the second best accuracy, it was selected to undergo the improvement.

In order to test E-(LP)², the system's enhanced version, our experiments were performed on English texts related to the task of seminar calls. More precisely, the CMU Seminar Announcements corpus which was used by Fabio Ciravegna for testing (LP)², is deployed. It will be briefly described in what follows.

6.2 Corpus Description

CMU Seminar Announcements is among the most popular corpus for testing Information Extraction systems. It was used in several works such as semantic annotation of semi-structured documents [36], hybridization between Rule-based and Statistical methods for general IE [37] and training of approximate CRF-based systems [38]. This corpus was created and labeled by Dayne Freitag [39] at Carnegie Mellon University. It consists of 485 files, each one represents an announcement giving details about an upcoming seminar.

CMU SA corpus contains different categories of concepts or named entities. In our experiments we focused on the Speaker concept, for which (LP)2 got a relatively low accuracy. The corpus contains 769 named entity belonging to this category. The information extraction task is to identify all Speaker mentions in each seminar announcement, by inserting the tag <Speaker> in the text.

In the following subsection, the procedure adopted to test E-(LP)² is outlined.

6.3 Experimental Settings

To identify all the instances of the Speaker concept, we divided the global CMU SA corpus into nearly equally-sized training and test corpus. The first one contains 243 files or announcements and the second one contains 242. The training corpus (TrC) is manually pre-annotated with tags <*speaker*> for each mention of the target concept while the test corpus (TeC) is not annotated. The number of Speaker instances mentioned in the TrC and the TeC is 335 and 434, respectively.

The goal of the experiment is to extract the Speaker names contained in the TeC. For this purpose, we started by training E-(LP)2 based on the TrC that is approximately constituted of a random half of the global corpus. The result of this step is a set of Best General Rules (BGR) [18] obtained after the IRs Generalization. Afterwards, E-(LP)2 is tested by applying the BGR on the test corpus. Then, the following usual metrics are calculated:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (2)$$

$$\text{F1} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \quad (3)$$

Where:

- TP (True Positive) is the number of extracted entities that are correct;
- FP (False Positive) is the number of extracted entities that are incorrect;
- FN (False Negative) is the number of entities that are correct but the system has not extracted.

The experiment is performed on a PC having the following configuration: 4 GB RAM, Core(TM) i3 and 2.50 GHz. The corpus was preprocessed using just Part-of-Speech tags and capitalization information as in [18].

6.4 Results and Comparison

Table 5 presents the system's enhanced version scores in terms of Precision, Recall and F-measure (F1), obtained for the Speaker concept under the task of seminar announcements. In Table 6, we compare these results with those obtained by (LP)².

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5: The Results Obtained by E-(LP)^2 on the CMU SA Corpus.

	Precision (%)	Recall (%)	F1 (%)
Speaker	90.5	89.8	90.21

Table 6: Comparison Between (LP)^2 [24] and E-(LP)^2 Results on the CMU SA Corpus Concerning the Concept Speaker.

	(LP)^2	E-(LP)^2	Difference
Precision (%)	87	90.59	3.59
Recall (%)	70	89.84	19.84
F1 (%)	77.60	**90.21**	**12.61**

As we can see in Table 6, the column *Difference* presents the difference between the values of Precision, Recall and F1 achieved by E-(LP)^2 and those obtained with (LP)^2. It shows that F1 score has increased by 12.5% thanks to our solution. Besides, Table 7 proves that E-(LP)^2 scores the best accuracy and outperforms other important Rule-learning and Statistical IE systems in the task, like RAPIER (+37.2%), TIPSI (+14.4%) and FastText (+6.6%).

Table 7: Comparison Between the Accuracy Obtained by E-(LP)^2, RAPIER [20], TIPSI [17] and FastText [27] on the CMU SA Corpus for the Speaker Concept.

	F1 (%)
E-(LP)^2	90.2
FastText	83.6
TIPSI	75.7
RAPIER	52.9

We would like to mention that FastText requires a specific format for the training and test corpus. Therefore, to test it on the CMU SA corpus, we defined two types of labels and transformed the text so that each Speaker instance will be contained in one line of the corpus file and preceded by the tag *__label__Speaker*. The other entities are preceded by the tag *__label__NotSpeaker*.

6.5 Discussion

E-(LP)^2 experimental results show an accuracy increase from 77.6% to 90.21% (Table 6) with respect to (LP)^2. Among the reasons that made the Recall increase by a quite large value (19.8%) is a basic tweak we applied to the extraction window. Actually, in order to solve the information loss problem mentioned in section III, we adjusted the extraction window to make E-(LP)^2 extract the positive example and *2*n* tokens to its neighborhood (*n* tokens before the opening tag and after the closing tag) while extracting the Initial Snippet. In this way, the system will include some words having a strong relatedness with the target concept and located after the positive example, like "*presents*" and "*talk*" in Example 1. Although this tweak increased the number of tokens in the IRs, it enabled E-(LP)^2 to generate from the latter new and efficient general rules that (LP)^2 wasn't able to induce. Hence, some Speaker instances that weren't covered by the system's initial version, were extracted. Therefore, FN (False Negative) decreased and the Recall reached a better score.

Another reason for the excellence of our experimental results relies in the use of OntoSA to remove from the IS some words having a low relatedness with the Speaker concept. Indeed, in (LP)^2, the presence of these words leads to the generation of some GRs which return erroneous annotations (False Positive). For example, the GR described in Table 8 inserts the tag *<Speaker>* before two successive words whose POS tags are NP, when they are followed by the token "*an*".

Table 8: Example of a GR Considered Among the Best Rules But Returning Incorrect Annotations on the Test Corpus.

Token index	Condition				Action (Tag)
	String	Lemma	POS tag	Case	
4			NP		Insert <Speaker> before Token 4
5			NP		
6	an				

This rule corresponds to the positive example in "*<Speaker> Katarzyna Klich </Speaker> an environmental leader*" and is considered among the best rules because it has a low error rate and identified correctly several instances in the training corpus. However, its application on the test corpus led to erroneous annotations like:

- *including a Central <Speaker> Pattern Generator </Speaker> an Adaptive Unit;*
- *<Speaker> No Longer </Speaker> an Oxymoron.*

Fortunately, E-(LP)^2, hasn't generated this GR because it removed the tokens *"an", "environmental" and "leader"* from the IS. Hence, our solution allowed to reduce the number of such GRs and thus decreased the number of False Positive examples (FP). This is in our view, the reason for the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

increase of the Precision score.

E-(LP)^2 has proved its superiority with respect to other IE systems like RAPIER, TIPSI and FastText (Table 7). The latter is famous and may obtain a better accuracy with very large datasets. However, it can't be applied directly on text corpus; they first have to be converted to a specific format as explained previously.

The proposed solution is generic. In other words, it can apply to other existing IE systems for rule learning like RAPIER and TIPSI. This aspect is novel because the state-of-the-art approaches are specific and dedicated to particular systems.

the lemmatization of the text tokens are performed via TreeTagger [40] (3.2.1) for Windows. Regarding the modelling of OntoSA ontology, we used Protégé 5.2.0 which is java-based, free and open source. The access to OntoSA is performed via Jena API (3.7.0).

We present in Figure 7 an interface allowing the administrator and the user to extract information from a selected text. To make the system generate the extraction rules from the pre-loaded training corpus, the user must click on "GENERATE RULES". Afterwards, he can load a text and click on "APPLY RULES" to get the latter annotated as depicted in Figure 8.

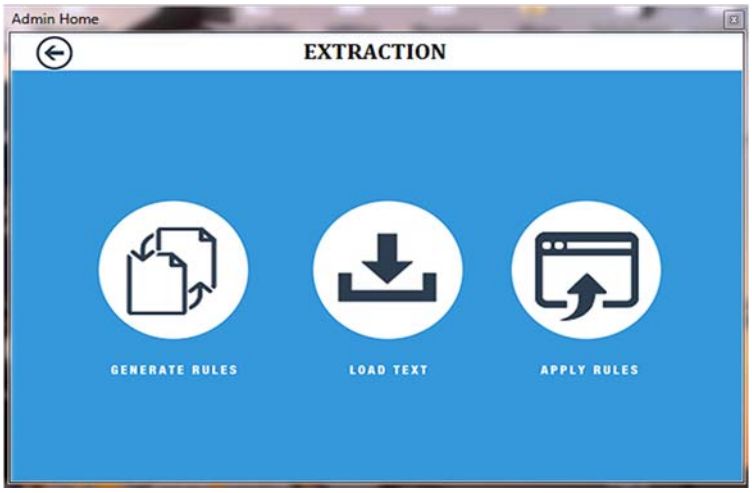

Figure 7: E-(LP)^2 Interface for Extracting Information

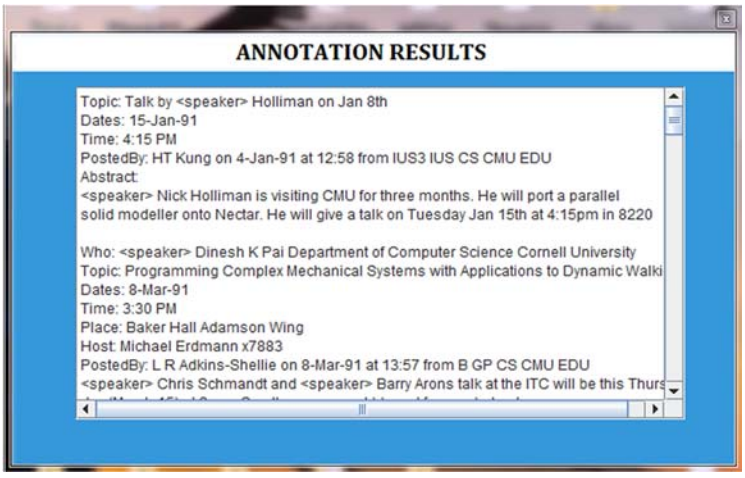

Figure 8: The Annotations Performed by E-(LP)^2 System

6.6 Validation

Our solution is implemented using the Eclipse IDE (4.6.1) for Java. The POS tagging and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

7. CONCLUSION AND FUTURE WORK

In this paper, we presented a solution for increasing the performance of Information Extraction systems that learn rules automatically. We resolved the main problem by improving the rule Generalization process based on a domain ontology to enable these systems to generate only the most likely relevant rules. Our experimental results demonstrate the effectiveness of E-(LP)^2 system, the enhanced version of (LP)^2 that was studied as a use case. Indeed, it reached a high accuracy compared to many state-of-the-art extractors. Besides, its accuracy increased by 12.5% with respect to (LP)^2, which means that E-(LP)^2 reduced the number of missing or erroneous instances returned by the system's initial version. Hence, our solution allows to provide more accurate information that better meets the user need. Its innovative and novel aspect lies in the fact that it is generic; it can apply to many Rule-learning systems like those named in this paper.

The extension of OntoSA presented in Section V, is among our future work goals. We aim to enrich the ontology with synonyms based on the large electronic semantic network WordNet [41]. This track will improve the accuracy even more, because it will help to cover additional true positive examples.

REFERENCES

[1] A. Ben Abacha and P. Zweigenbaum, "Une étude comparative empirique sur la reconnaissance des entités médicales," *Traitement Automatique des Langues,* Vol. 53, No. 1, 2012, pp. 39-68.

[2] F. Jebbor and L. Benhlima, "Towards a Conversational Question Answering System," *Lecture Notes in Electrical Engineering,* Vol. 380, Springer, cham, 2016, pp. 307-315.

[3] J. Piskorski and R. Yangarber, "Information extraction: past, present and future," *Multi-source, Multilingual Information Extraction and Summarization*, Springer, Berlin Heidelberg, 2013, pp. 23-49.

[4] CH. Chang, HM. Chuang, CY. Huang et al., "Enhancing POI search on maps via online address extraction and associated information segmentation," *Applied Intelligence,* Vol. 44, No. 3, Oct. 2016, pp. 539-556.

[5] J. Sangeetha and S. Jothilakshmi, "Speech translation system for english to dravidian languages," *Applied Intelligence,* Vol. 46, No. 3, Sep. 2017, pp. 534-550.

[6] J. Jiang, "Information extraction from text," *Mining Text Data*, Springer, Boston, MA, 2012, pp. 11-41.

[7] J. Cheng, X. Zhang, P. Li et al., "Exploring sentiment parsing of microblogging texts for opinion polling on chinese public figures," *Applied Intelligence,* Vol. 45, No. 2, Mar. 2016, pp. 429-442.

[8] L. Chiticariu, Y. Li and F. R. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems," *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Washington, 2013, pp. 827–832.

[9] J. Strotgen, J. Zell and M. Gertz, "HeidelTime: Tuning english and developing spanish resources for tempeval-3," *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SemEval 2013)*, Georgia, 2013, pp. 15–19.

[10] N. Chambers, "NavyTime: Event and time ordering from raw text," *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SemEval 2013)*, Georgia, 2013, pp. 73–77.

[11] P. S. Jacobs, Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval, Psychology Press, New York, 2014.

[12] Y. Li, E. Kim and M. A. Touchette et al., "Vinery: A visual ide for information extraction," *Proceedings of the 41st International Conference on Very Large Data Bases*, Kohala Coast, 2015, pp. 1948-1951.

[13] S. Sarawagi, "Information Extraction," *Foundations and Trends in Databases*, Vol. 1, No. 3, 2008, pp. 261-377.

[14] R. Grishman and J. Sterling, "New York University: Description of the PROTEUS system as used for MUC-3," *Proceedings of the 3rd Message Understanding Conference*, Baltimore, 1993, pp. 181–194.

[15] H. Cunningham, D. Maynard and V. Tablan, "JAPE: a Java annotation patterns engine," Research memo, University of Sheffield, UK, 2000.

[16] C. C. Aggarwal and C. X. ZHAI, Mining text data, Springer-Verlag, New York, 2012.

[17] Zhang K., Li J., Hong M., Yan X., Song Q. (2014) A Semantics Enabled Intelligent Semi-structured Document Processor. In: Yuan Y., Wu X., Lu Y. (eds) Trustworthy Computing and Services. ISCTCS 2013. Communications in Computer and Information Science, vol 426. Springer, Berlin, Heidelberg

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[18] F. Ciravegna, "(LP)²: Rule Induction for Information Extraction Using Linguistic Constraints," Technical Report CS-03-07, University of Sheffield, UK, 2003.

[19] E. Minkov, "Event extraction using structured learning and rich domain knowledge: Application across domains and data sources," *ACM Transactions on Intelligent Systems and Technology,* Vol. 7, No. 2, Jan. 2016, pp. 1-34.

[20] M. E. Califf and R. J. Mooney, "Bottom-up relational learning of pattern matching rules for information extraction," *Journal of Machine Learning Research*, Vol. 4, Jun. 2003, pp. 177-210.

[21] D. Freitag and N. Kushmerick, "Boosted Wrapper Induction," *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, California, 2000, pp. 577-583.

[22] Chiticariu L, Li Y, Reiss FR (2013) Rule-based information extraction is dead! long live rule-based information extraction systems, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. The Association of Computational Linguistics, Washington, Jun. 2003, pp 827–832.

[23] J. Struyf and H. Blockeel, "Relational learning," *Encyclopedia of Machine Learning*, Springer, Boston, pp. 37-45, 2011.

[24] F. Ciravegna, "Learning to Tag for Information Extraction from Text," *Proceedings of the 14th European Conference on Artificial Intelligence*, Amsterdam, 2000.

[25] F. Ciravegna, "Adaptive information extraction from text by rule induction and generalization," *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, San Francisco, 2001, pp. 1251-1256.

[26] T. Mikolov, K. Chen, G. Corrado et al. "Efficient estimation of word representations in vector space," *Proceedings of the International Conference on Learning Representations*, Arizona, 2013.

[27] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, "Bag of tricks for efficient text classification," *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Spain, 2017, pp. 427-431.
International Conference on Machine Learning, San Francisco, 1998, pp. 161-169.

[28] S. K. Siencnik, "Adapting word2vec to named entity recognition," *Proceedings of the 20th nordic conference of computational linguistics*, Lithuania, 2015, pp. 239-243.

[29] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, Vol. 5, 2017, pp. 135–146.

[30] H. Ehrenberg and D. Iter, "Ensembling Insights for Baseline Text Models," https://web.stanford.edu/class/cs224n/reports/2758157.pdf.

[31] E. Dusserre and M. Padro, "Bigger does not mean better! We prefer specificity," *Proceedings of the 12th international conference on computational semantics*, France, 2017.

[32] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review,* Vol. 11, No. 2, Jun. 1996, pp. 93-136.

[33] V. Sazonau, U. Sattler and G. Brown, "General terminology induction in OWL," in *Lecture Notes in Computer Science*, Springer, Cham, 2015, pp. 533-550.

[34] K. E. Campbell, S. P. Cohn, C. G. Chute, G. D. Rennels and E. H. Shortliffe, "Gálapagos: Computer-based support for evolution of a convergent medical terminology," *Proceedings of the AMIA Fall Symposium*, Washington, 1996, pp. 269–273.

[35] M. A. Musen, "The Protégé project: A look back and a look forward," *AI Matters,* Vol. 1, No. 4, Jun. 2015, pp. 4-12.

[36] K. Zhang, J. Li, M. Hong, X. Yan and Q. Song, "A Semantics Enabled Intelligent Semi-structured Document Processor," *Communications in Computer and Information Science*, Springer, Berlin, 2014, pp. 328-344.

[37] M. B. Grap, "A Hybrid Approach to General Information Extraction," M.S. thesis, California Polytechnic San Luis Obispo, US, 2015.

[38] V. Stoyanov and J. Eisner, "Minimum-risk training of approximate CRF-based NLP systems," *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, 2012, pp. 120-130.

[39] D. Freitag, "Multistrategy learning for information extraction," *Proceedings of the 15th*

[40] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," *Proceedings of the International Conference on New Methods*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in Language Processing, Manchester, 1994, pp. 44-49.

[41] C. Fellbaum, "WordNet," in *Theory and Applications of Ontology: Computer Applications*, Springer, Dordrecht, 2010, pp. 231-243.

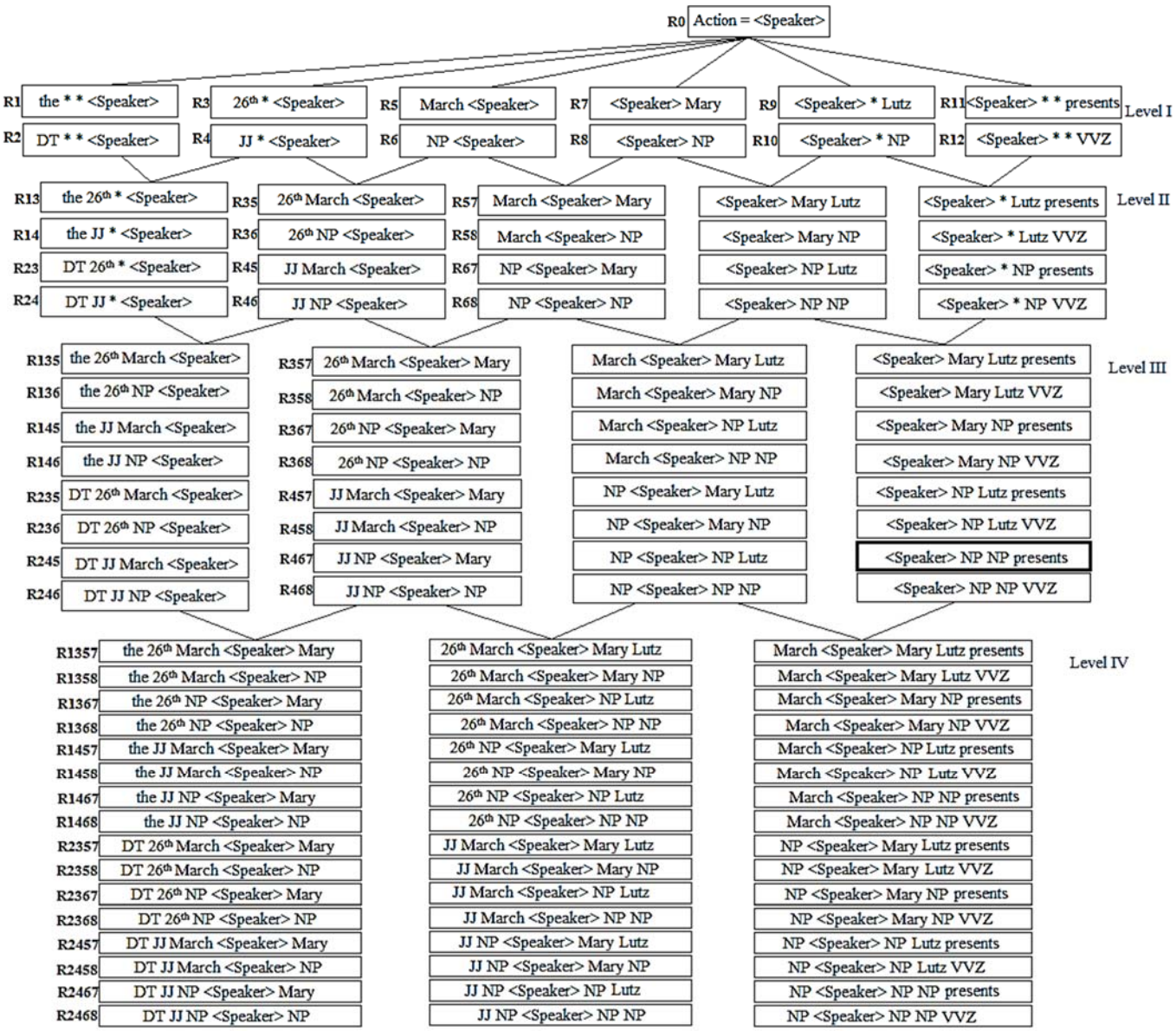

Figure 2: Extract From the Generalization Tree Corresponding to the IR in Table 1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

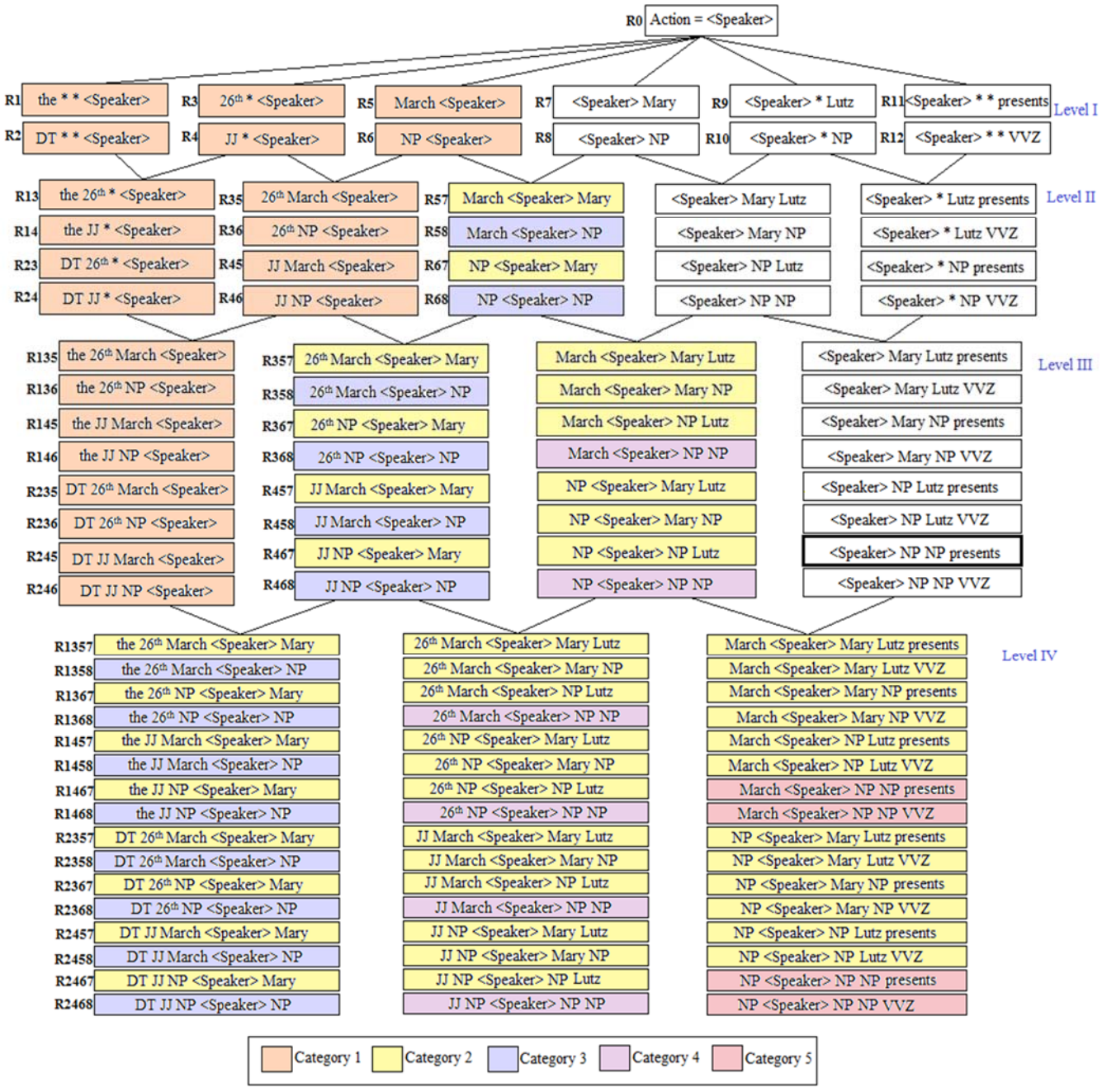

Figure 6: Categorization of the Irrelevant General Rules that Weren't Produced After the Improvement