Exploiting Machine Learning Techniques to Build an Event Extraction System for Portuguese and Spanish

Hristo Tanev
JRC, Ispra, Italy
hristo.tanev@ext.jrc.ec.europa.eu

Vanni Zavarella
JRC, Ispra, Italy
vanni.zavarella@ext.jrc.ec.europa.eu

Jens Linge
JRC, Ispra, Italy
jens.linge@jrc.ec.europa.eu

Mijail Kabadjov
JRC, Ispra, Italy
mijail.kabadjov@jrc.ec.europa.eu

Jakub Piskorski
Polish Academy of Sciences
Jakub.Piskorski@ipipan.waw.pl

Martin Atkinson
JRC, Ispra, Italy
martin.atkinson@jrc.ec.europa.eu

Ralf Steinberger
JRC, Ispra, Italy
ralf.steinberger@jrc.ec.europa.eu

November 22, 2009

Abstract

We describe a multilingual methodology for adapting an event extraction system to new languages. The methodology is based on highly multilingual domain-specific grammars and exploits weakly supervised machine learning algorithms for lexical acquisition. We adapted an already existing event extraction system for the domain of conflicts and crises to Portuguese and Spanish languages. The results are encouraging and demonstrate the effectiveness of our approach.

1 Introduction

We present a multilingual methodology for building event extraction systems and describe its application for the Portuguese and Spanish languages. Formally, the task of event extraction is to automatically identify events in free text and to derive detailed information about them, ideally identifying *Who did what to whom, when, with what methods (instruments), where and why.* Automatically extracting events is a higher-level information extraction (IE) task (Appelt, 1999) which is not trivial due to the complexity of natural language and due to the fact that, in news, a full event description is usually scattered over several sentences and articles. In particular, event extraction relies on identifying named entities and relations between them. The research on automatic event extraction was pushed forward by the DARPA-initiated Message Understanding Conferences[1] and by the ACE (Automatic Content Extraction)[2] programme. Although, a considerable amount of work on automatic extraction of events has been reported, it still appears to be a lesser studied area in comparison to the somewhat easier tasks of named-entity and relation extraction.

First attempts to larger-scale event extraction systems were reported a decade ago, e.g., in (Aone and Santacruz, 2000). Some examples of the current functionality and capabilities of event extraction technology dealing with identification of disease outbreaks, conflict incidents and other crisis-related events are given in (Grishman and Yangarber, 2002),(Grishman and Yangarber, 2003), (King and Lowe, 2003), (Naughton and Carthy, 2006), (Ji and Grishman, 2008), (Yangarber, Rauramo, and Huttunen, 2005) and (Wagner and Baker, 2006).

We have created a multilingual event extraction system NEXUS, which is part of the Europe Media Monitor family of applications (EMM) (Steinberger, Pouliquen, and van der Goot, 2009). EMM performs automatic real-time gathering and analysis of online news in 45 languages. NEXUS aims at identifying vio-

[1] http://en.wikipedia.org/wiki/Message_Understanding_Conference

[2] ACE - http://projects.ldc.upenn.edu/ace

lent events, man made and natural disasters and humanitarian crises, in news reports. The information about such events is extremely important for better crisis management and for developing warning systems which detect precursors for threats in the fields of disaster and conflict.

Crucial information for all these events are the number and the description of the victims. Additionally, analysis of humanitarian crises requires identification of the number of the displaced and homeless people; analysis of the violent events requires identification of the weapons and the perpetrators.

Currently, NEXUS can handle 4 languages - English, French, Italian, and Russian. Within the EMM project, we aim at global monitoring of crisis and conflict events: at the same time, we also try to detect events with only national or local relevance. In this view, we decided to adapt Nexus to Portuguese and Spanish language so as to extend the coverage of our system to Latin American and African areas.

The architecture and the algorithms implemented in NEXUS are highly language-independent. The system involves the use of language-specific dictionaries and extraction grammars, which are plugged in as external resources; therefore, adding a new language to the system is possible without modifying the system itself. Moreover, the domain-specific grammars, which we use to extract event-specific entities, contain very few references to concrete words. Therefore, a grammar for one language can be reused without significant changes for another language, especially if they belong to the same language family. In our development cycle, we adapted an Italian grammar to other members of the Romance language family, namely French, Spanish and Portuguese.

In order to adapt our event extraction system to a new language, we adopted a multilingual methodology which is based on two semi-supervised machine learning algorithms and highly language-independent domain-specific grammars. Using this methodology, we were able to build event extraction systems for the Portuguese and Spanish languages with promising performances, which proved the viability of our approach.

In section 2 we outline the architecture of NEXUS and its integration within the European Media Monitor system; section 3 outlines the extraction grammar as it was adapted for the Portuguese and Spanish; then section 4 describes the machine learning algorithms we exploit and finally we present experiments and evaluation.

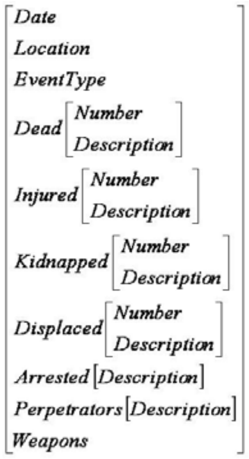


Figure 1: The output structure of the event extraction system

2 EMM and NEXUS

Europe Media Monitor (EMM) is an ongoing project, whose main outcome is a multilingual news gathering and analysis system which works for 41 languages, (Steinberger, Pouliquen, and van der Goot, 2009).

The NEXUS event extraction system takes on its input the information provided by other EMM modules, integrates it after performing validation and merging, in order to extract event report summaries. Before the proper event extraction process can proceed, news articles are gathered by dedicated software for media monitoring, that receives 90000 news articles from 2200 news sources in 41 languages each day. Next, the articles are grouped into news clusters according to content similarity. Subsequently, each cluster is geo-located.

For each such a cluster NEXUS tries to detect and extract only the main event by analyzing the title and first sentence of all of the articles in the cluster. For each detected violent and disaster event NEXUS produces a frame, whose main slots are shown in Figure 1.

In Figure 2, a sketch of the entire event extraction processing chain is shown. First, the full news article are scanned by EMM modules in order to identify entities and locations which are inserted as meta-data. These entities are typically separate from the ones deployed in the event extraction process proper. Next the articles are clustered and then geo-located according to extracted meta-data. Each article in the cluster is then linguistically preprocessed in order to produce a more abstract representation of its text. This encompasses the following steps: fine-grained tokenization, sentence splitting, domain-specific dictionary look-up (i.e. matching of key

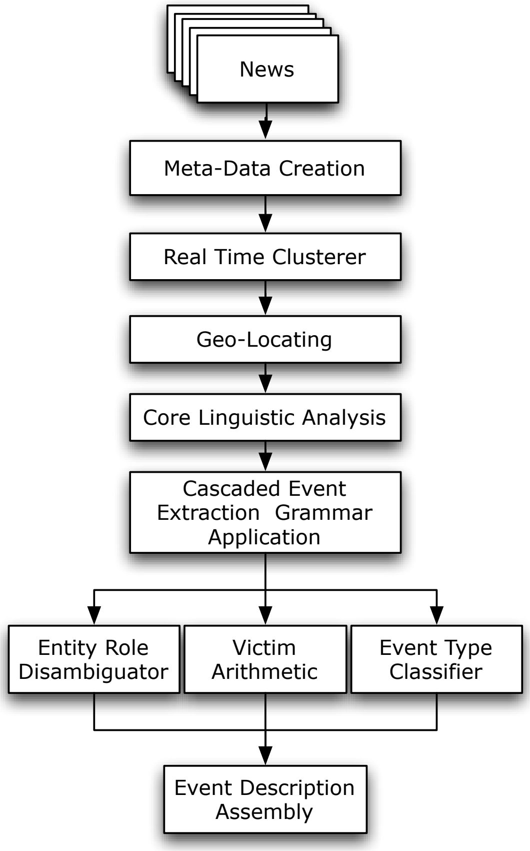


Figure 2: Event Extraction processing chain

terms indicating numbers, quantifiers, person titles, unnamed person groups like *civilians*, *policemen* and *Shiite*), and finally morphological analysis, simply consisting of lexicon look-up on large domain-independent morphological dictionaries. The aforementioned tasks are accomplished by CORLEONE (Core Linguistic Entity Online Extraction), our in-house core linguistic engine (Piskorski, 2008). Once the linguistic preprocessing is complete, a cascade of extraction grammars is applied on each article in order to identify phrases reporting about victims and entities and their participation in the event. For example, phrases like *”matou seis civis”* are parsed by the grammar cascade, extracting the *seis civis* as victims.

The news clusters contain reports from different news sources about the same fact. This redundancy mitigates the impact on system performance of linguistic phenomena which are hard to tackle, such as anaphora, ellipsis and long-distance dependency. Consequently, the system can process the first sentence and the title of each article, where the main facts are summarized in simple syntax (Bell, 1991), without significant loss in coverage.

On the other hand, contradictory information on the same story may occur at the cluster level; consequently, the last processing steps consist of cross-article information fusion in order to produce event descriptions. Namely, Nexus aggregates and validates information extracted locally from each single article in the same cluster. This process encompasses mainly three tasks, entity role disambiguation (as a result of extraction pattern application the same entity might be assigned different roles), victim counting and event type classification. An example of the system output as geolocated in a Google Map interface is shown in Figure 3.

3 Outline of the Extraction Grammars

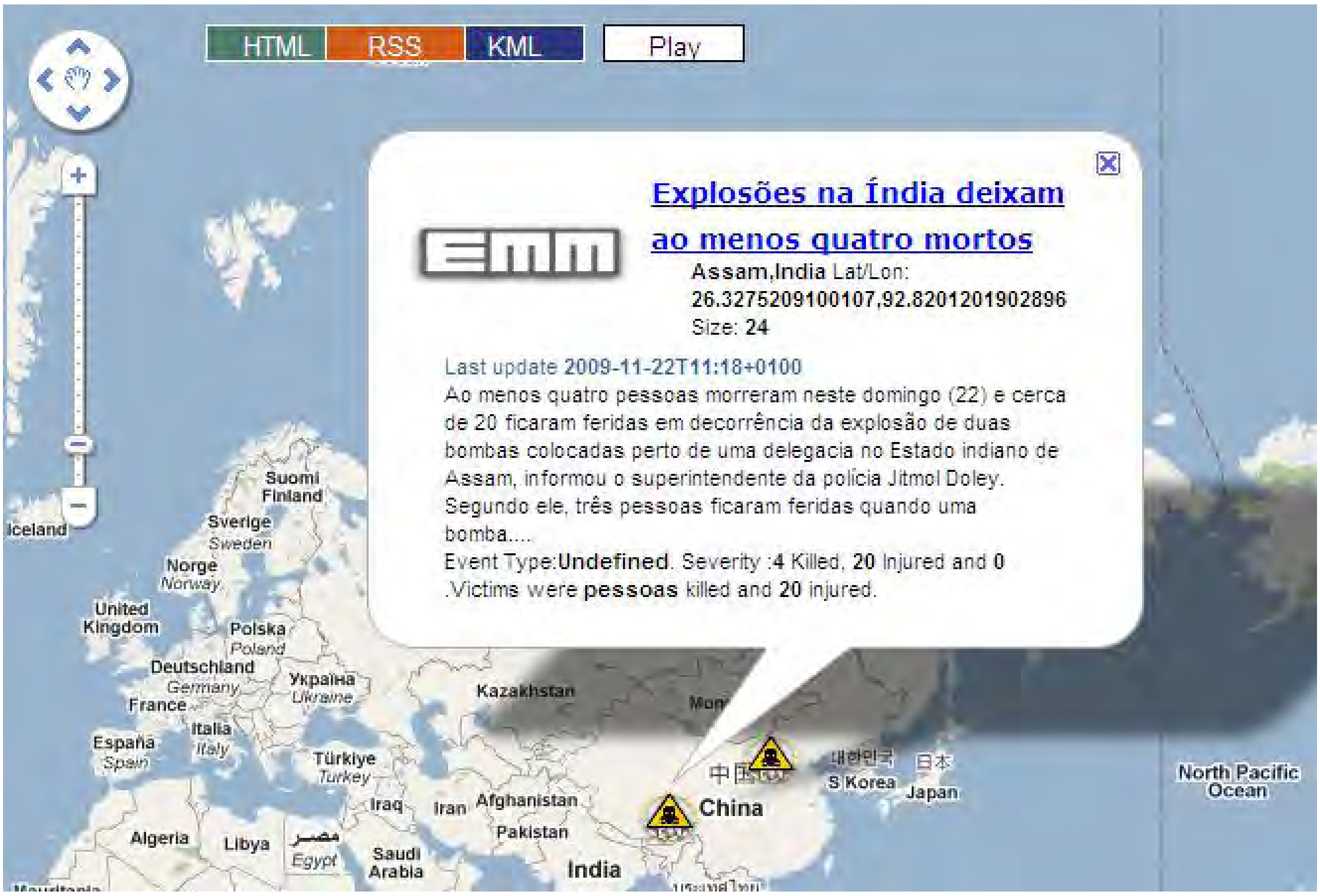
The role of the grammars deployed in NEXUS is the recognition of phrases which introduce events participants. For example, in the text

Soldados israelenses matam palestino de 14 anos

the grammar should extract the phrase *Soldados israelenses* and assign to it the semantic role *perpetrator*, while the phrase *palestino de 14 anos* should be extracted as *dead victim* description. The extraction process is performed by devising a multi-layer grammar cascade in the EXPRESS formalism (Piskorski, 2007). EXPRESS is a finite state-based grammar formalism and pattern matching engine developed in-house which proved quite fast and efficient in real-time text processing.

3.1 Extraction Pattern Specification Language

An EXPRESS grammar consists of a cascade of pattern-action rules. The left-hand side (LHS) of a rule (the recognition part) is a regular expression over flat feature structures (FFS), i.e., non-recursive typed feature structures (TFS) without structure sharing, where features are string-valued and types are not hierarchically ordered (differening in this from traditional unification-based grammar formalisms). The right-hand side (RHS) of a rule (action part) consists of a list of FFS, which is returned in case the LHS pattern is matched. Variables can be associated to the string-valued attributes on the LHS of a rule in order to allow information transport into the RHS. Further, functional operators are allowed in the RHSs in order to form output slot values by string processing operations or specify constraints in the form of boolean-valued predicates. Rules can be associated with multiple ac-



[Explosões na Índia deixam ao menos quatro mortos]
Assam,India Lat/Lon:
26.3275209100107,92.8201201902896
Size: 24

Last update 2009-11-22T11:18+0100
Ao menos quatro pessoas morreram neste domingo (22) e cerca de 20 ficaram feridas em decorrência da explosão de duas bombas colocadas perto de uma delegacia no Estado indiano de Assam, informou o superintendente da polícia Jitmol Doley. Segundo ele, três pessoas ficaram feridas quando uma bomba....
Event Type:Undefined. Severity :4 Killed, 20 Injured and 0 .Victims were pessoas killed and 20 injured.

Figure 3: A sample output of the event extraction system as shown in Google Map interface.

tions, i.e., producing multiple annotations (possibly nested) for a given text fragment. Finally, arbitrary processing resources can be integrated at any level of the grammar cascade. In our case, CORLEONE text processing modules are deployed. For more details on the EXPRESS formalism and its processing performance refer to (Piskorski, 2007).

3.2 Person and entity recognition

The lower levels of the grammar cascade contain patterns for recognition of named entities (e.g., person names), numbers, quantifiers, simple chunks representing unnamed person groups (e.g., *cinco policiais, milhares de portugueses, casi la mitad de los soldados extranjeros*); moreover, appositive and coordinated phrase composition are covered. On Figure 4 is presented a simplified example of a rule for detection of an unnamed person entity noun phrase such as *um jovem militante*. The rule matches a sequence consisting of an optional article, followed by a nationality noun, preceded and/or followed in its turn by an optional modifying adjective - so as to deal with relatively free position of modifiers in Romance language noun phrases. It produces a singular *person_group* type structure.

Expressions like *adjective* or *gazetteer* at the front of FFS's make reference to one of the output types of CORLEONE modules which are available at the level of the grammar cascade where the rule appears - in this case, morphological and domain-specific lexicon look-up. The symbol "&" links the name of the FFS's type with a list of constraints (in the form of attribute-value pairs) to be fulfilled, such as on grammatical number, morphological subtype, gender and so on.

Notice that the NAME value of the output structure is created by accessing the variables*#name0*, *#name1* and *#name2* on the LHS and concatenating them by calling a functional operator, while GENDER attribute value is read directly from grammatical gender of article and/or modifiers , if there are any; Gender agreement is then enforced by the string equality predicate *IsEqual()*.

Notice how such a simple rule would run almost identically for Portuguese, Spanish and other Romance languages, provided that suitable lexicons for domain specific categories such as *person* above, or for general grammatical categories (like *adjective*) were plugged into the system.

```
person_entity_sg :>
         ( determiner & [TYPE:"article", NUMBER:"sg", GENDER:#g0]?
           adjective & [NUMBER: "sg",GENDER:#g1, SURFACE: #name0]?
           gazetteer & [GTYPE: "person",NUMBER: "sg", GENDER:#g, SURFACE: #name1]
           adjective & [NUMBER: "sg", GENDER:#g2, SURFACE: #name2]? ):name
-> name: person_group & [NAME: #name, TYPE: "U_PER", GENDER:#g,
          AMOUNT:"1", NUMBER:"sg", RULE: "person_entity_sg"]
                & #name:=Concatenate(#name0,#name1,#name2)
                & IsEqual(#g0,#g1,#g,#g2).
```

Figure 4: Rule for detection of phrases referring to persons

More word token-level rules are also deployed in order to detect named person expressions in text such as *Doutor Eduardo R. Souza* etc. Finally, composition of *person* and *person_group* types into larger phrases is captured by higher level rules like the one shown on Figure 5, which matches appositional phrases like: *dois jovens de 20 anos, Paulo Souza e Gilberto Fernandez.*

Here, the constraints *IsNotUnspecified()* on the AMOUNT attributes are used to enforce the matching of person name coordinations and their appositive descriptions, while excluding underspecified quantifiers such as *algunas personas.*

All in all, person and entity recognition grammar is abstracting from surface forms and relies rather on: a) a number of fine-grained token classes (e.g., word-with-hyphen, word-with-apostrophe, all-capital-letters), which are to a large extent language-independent; b) person name and partial noun phrase syntactic structure; c) lexical resources for the target language. Because b) has low variation over Romance languages and only limited differences with respect to English, the process of person grammar porting onto Portuguese and Spanish was relatively straightforward and required limited level of linguistic expertise. Therefore, size and complexity of the grammars could be kept relatively low and the bulk of grammar development was mostly on providing suitable lexical resources.

We make use of two types of lexica:

1. morphological dictionaries for Portuguese and Spanish;

2. domain-specific lexicons, listing a number of (possibly multiword) expressions, subcategorized into semantic classes relevant for the domain of violent and disaster events, with limited or no linguistic annotation; classes range from person names, quantifying expressions (like *pelo menos dez*), through weapons and person positions (e.g. *grevistas*, *emigrante*, *passageiros*, *niños*, *mujer*).

As for the morphological dictionary, we make use of LABEL-LEX-sw electronic lexicon (Samuel et al., 1995), listing about 1M simple Portuguese wordforms. For Spanish we used MULTEXT, which encompasses 510K wordforms. It is important to note that we use MULTEXT (Erjavec, 2004) in order to perform morphological look-up, mainly due to the fact that MULTEXT tags are uniform for all languages. These resources are noticeably large, nonetheless we do not frequently make reference to abstract POS classes like nouns in our person recognition grammars as we noticed this highly exposes to the risk of overgeneralization and reduced accuracy. Consequently, we estimate a first potential bottleneck of the extraction process to be on the coverage and accuracy of domain-specific semantic classes; we will show in the next section how we generated and extended these resources.

3.3 Event triggering patterns

Prior to person recognition grammar application, event triggering linear patterns are matched on text for extraction of partial information on event roles, such as actors, victims, etc. These patterns are similar in spirit to the ones used in AutoSlog (Riloff, 1993). We use 1/2-slot surface level patterns like the following English and Portuguese samples, where role assignments are shown in brackets:

```
<DEAD> was shot by <PERPETRATOR>
police nabbes <ARRESTED>
<KIDNAPPED> has been taken hostage
<WOUNDED> was found injured
raptou <KIDNAPPED>
<DEAD> foram mortas
```

Note that the role slots (in brackets) can be filled by phrases referring to persons or person groups.

Patterns are stored in a domain-specific lexicon, each one associated with a type indicating the position of the pattern with respect to the slot to be filled (left or right), the event-specific semantic role assigned to the entity filling

```
person_group_apposition_rule :>
 (person_group & [NAME:#description, NUMBER:"p", AMOUNT:#amount1]
               token & [SURFACE: "," #com1]?
   person_group & [NAME:#name,NUMBER:"p",AMOUNT:#amount2]
   token & [SURFACE: ","]? ):noun_phrase
-> noun_phrase: person_group & [NAME:#final, AMOUNT:#amount1,
NUMBER:"p",RULE:"person_group_apposition_rule"]
& #final := ConcWithBlanks(#description,#com1,#name)
& IsEqual(#amount1,#amount2)
& IsNotUnspecified(#amount1)
& IsNotUnspecified(#amount2).
```

Figure 5: Rule for detection apposition phrases

the slot (e.g., DEAD, PERPETRATOR) and the grammatical number of the phrase which may fill the slot. For instance, the following represents the encoding for the surface pattern ”foi sequestrada” detecting a kidnapped person:

```
foi sequestrada [
   TYPE: right-context-sg-and-pl,
   SURFACE: "foi sequestrada",
   SLOTTYPE: KIDNAPPED]
```

Through such a compact encoding, linear patterns can be then combined with detected person and person group entities at the top level of the grammar cascade via extraction rules like the simplified sample on Figure 6, which detects an Injuring event, extracting description and number of the victims.

These rules are meant to model simple domain-specific language constructions describing events, with extraction patterns being linearly non-overlapped with person phrases. For English language, strict word order and relatively simple morphology made such a surface level approach perform well in terms of both precision and recall (Piskorski, Tanev, and Wennerberg, 2007).

4 Semi-supervised resource acquisition

An important element in our approach is the usage of weakly supervised machine learning tools to acquire the language-specific resources which the system needs for processing the new languages. Namely, we use a news cluster based method for pattern learning, described in (Piskorski, Tanev, and Wennerberg, 2007) and we use a new weakly supervised approach (based on (Tanev and Magnini, 2006)) for learning of semantic categories, such as nouns, referring to people and weapons.

4.1 Ontopopulis - a system for learning of semantic categories

For each language our event extraction system should have among the other resources a list of phrases belonging to two semantic categories: weapons and persons. Event extraction uses this information in order to recognize entities mentioned in the articles (e.g. weapons) and also to parse noun phrases referring to specific semantic classes, such as people. We also learned several semantic categories which were used for event classification: vehicles, infrastructural objects, crimes, edge weapons and politicians.

There are different approaches for term extraction and categorization, however we have specific settings: First, we lack annotated data. On the other hand, we had available an unannotated corpus of Portuguese and Spanish news. Finally, we only had to learn few semantic classes. Considering this, we found quite relevant the weakly supervised term classification approach described in (Tanev and Magnini, 2006). Based on it and on its extention, presented by (Shi, Sun, and Che, 2007), we created our own term extraction and classification system - Ontopopulis.

Ontopopulis takes on its input a set of seed terms for each semantic category under consideration and an unannotated corpus of news articles. For example, for the category *weapons* in Portuguese we used terms like *arma branca*, *navalha*, *metralhadora*, etc. and for the category *persons*: *soldado*, *mulher*, *governador*, etc. The system performs two learning stages - Feature Extraction and Term Extraction:

4.1.1 Feature extraction and weighting

For each category (e.g. *weapons*), we consider as a context feature each uni-gram or bi-gram n which co-occurs at least 3 times in the corpus with any of the seed terms from this category (we have co-occurrence only when n is adjacent to a seed term on the left or on the right). The feature

```
injury-event :>
       ((person-group & [NAME: #name1, NUMBER: #num1]):injured1
         gazetteer & [POS: "conjuntion"]
        (person-group & [NAME: #name2, NUMBER: #num2]):injured2
         injured-phrase & [FORM: "passive"]
        ):event
-> injured1: victim & [NAME: #name1, NUMBER: #num1],
     injured2: victim & [NAME: #name2, NUMBER: #num2],
     event: injury & [VICTIM: #name, NUMBER: #count],
   & #name = Concatenate(#name1," & ",#name2)
   & #count = EstimateNumber(#num1," ",#num2).
```

Figure 6: Rule for detection of injury events

can not be composed only of stop words; we also do not consider words beginning with capitalized letters and numbers.

For each such a context feature n and a semantic category cat we calculate the score:

$$score(n, cat) = \sum_{st \in seeds(cat)} PMI(n, st)$$

where $seeds(cat)$ are the seeds terms of the category cat and $PMI(n, st)$ is the pointwise mutual information which shows the co-occurrence between the feature n and the seed term st.

At the end of this learning phase the user performs manual feature selection from a list of 250 best scored features, suggested by the system. This step guarantees high quality features which is very important for the accuracy of the final results. For example, some of the top ranking learned and approved features for *weapons* in our experiments are: *tiro de W*, *golpes de W*, *armado com W*, *ataque com W*, here W stands for the position where the weapon-terms should appear. Here are some examples of extracted features about for the class *vehicle*: *acidente com um V*, *bordo de um V*, *passageiros do V*.

4.1.2 Term extraction and weighting

The term extraction and learning stage takes the features, which were learned and manually selected for each category in the previous stage and extracts as candidate terms uni-grams and bigrams, which tend to co-occur with these features and which do not contain stop words, numbers or capitalized letters. Weighting of the candidate terms was carried out with the view to optimize the efficiency of the calculations. For this reason, we avoid to obtain the frequency of each candidate term in the corpus and we rather calculate the term feature vector in a non-standard way. It would be statistically more correct to use as a feature weight the pointwise mutual information between the term and the feature. However, this would require to collect statistics about the term frequency, which will decrease the algorithm speed.

We weight the term candidates, using the following algorithm:

1. For each category C we define a feature space, whose dimensions are only the features selected for this category
2. For each category C we define a *category feature vector* $\overrightarrow{C} = (wf_1, wf_2, wf_3, ..., wf_{nc})$ where wf_i are the weights of the category features, calculated as $wf_i = score(n_i, C)$, where n_i is the n-gram used as ith feature in our model; $score(n_i, C)$ is calculated with the pointwise-mutual-information based formula presented in the previous subsection.
3. We normalize each *category feature vector* $\overrightarrow{C}$ by dividing its coordinates with its length and obtain $norm(\overrightarrow{C})$.
4. Then, for each candidate term t for the category C we define a term feature vector $\overrightarrow{t_C} = (wt_1, wt_2, ..., wt_{nc})$ where $wt_i = \frac{ft_i}{ft_i+3}$, ft_i is the frequence with which the candidate term t appears with feature i.
5. Finally the weigth for each candidate term t for a category C is defined as a scalar product in the vector space defined for the category C, multiplied by the square root of the number of the non-zero features of the term feature vector: $weigth(t, C) = \overrightarrow{t_C}.norm(\overrightarrow{C}).\sqrt{NNZF(t_C)}$, where $NNZF$ is the number of the features with non zero weight.

Finally, the system orders the term candidates for each category by decreasing weight and filters out terms with a weight under a certain threshold. Then, the term list is given to the user for manual cleaning.

4.2 Learning linear patterns

In order to acquire the linear patterns for extraction of victims, perpetrators and arrested people, we implemented an iterative pattern acquisition algorithm, whose output is validated by a human moderator on each step. This algorithm was originally suggested by (Piskorski, Tanev, and Wennerberg, 2007). Their automatic approach takes on the input an annotated corpus and learns event specific templates. We modified the approach in such a way that it takes on its input a small set of seed patterns and a corpus without annotations. Then, as a first step we annotate the corpus using these patterns and then we run the original algorithm Here are the basic steps of the pattern learning algorithm:

1. For a specific role like dead victim, injured victim, perpetartor, etc. the user provides a small set of seed patterns. For example, let's consider the role dead victim and the small set of seed patterns: [PERSON] "mortas", [PERSON] "mortos", where [PERSON] matches any person description. We use the person recognition grammar and the semi-automatically learned list of person terms (see the previous sub-section), in order to extract phrases which refer to people, e.g. "cinco pessoas".

2. Annotate a corpus of news clusters, using these patterns. For example, if the text "cinco pessoas mortas" appears , the phrase "cinco pessoas" will be annotated as dead victim.

3. Propagate annotation inside the news clusters. At this step, if in a news cluster there is an annotated phrase, such as "cinco pessoas", then all the occurrences of this phrase inside the same cluster will be annotated with the same semantic role, e.g. dead victim. The assumption behind this step is that all the articles in a news cluster report about the same event, therefore equal phrases refer to the same entity which usually appears in the same semantic role across the whole cluster.

4. Learn automatically linear extraction patterns from the left and right contexts of the annotated phrases. For example, the phrase "cinco pessoas" and other annotated ones may appear systematically in phrases like"mata" [PERSON] as a result of the annotation propagation. As a consequence, such patterns will be added to the list of the learned ones. An entropy-based pattern extraction algorithm was used to perform this stage of the learning process (see (Piskorski, Tanev, and Wennerberg, 2007) for detailed description).

5. Manually filter out low quality patterns.

6. If the user estimates that the list of patterns is good enough, terminate.
   Otherwise, go to step 2.

We used successfully this algorithm for different languages, including the Portuguese and Spanish. Rarely, it was necessary to run more than two iterations. This approach facilitates the adaptation of the event extraction system to new languages by significantly decreasing the human efforts necessary to create language specific pattern libraries. Moreover, the algorithm does not need an annotated corpus. The human efforts are concentrated in the final step of each iteration, where the user is required to clean the output list of patterns, which in general requires less efforts than annotating a corpus. (We also experimented with manual corpus annotation and consequent pattern learning, however we found out that this approach is slower than the one presented here).

5 Experiments and Evaluation

We tested our methodology for Spanish and Portuguese. For each language we performed a series of resource-creation steps, which enabled NEXUS to extract event reports in the corresponding language:

1. Adapt the person recognition grammar from Italian

2. Run Ontopopulis to learn a dictionary of persons, weapons and other categories

3. Manually validate and clean the output of Ontopopulis

4. Create manually a small list of closed-class words and multiwords, such as quantifiers

5. Run the pattern learning algorithm for each of the following semantic roles: dead, wounded, kidnapped, perpetrator, and arrested

6. Manually clean the output of the pattern learning algorithm

5.1 Evaluation of Ontopopulis

The main purpose of the experiments was to evaluate the application of our methodology to Portuguese and Spanish. In this clue, there are two important parameters which can be used to estimate quantitatively the outcome of our experiments: First, the accuracy of Ontopopulis and

	person	weapon	politician	vehicle	watercraft	edged weapon	crime	building
seed terms	48	26	46	135	28	20	33	73
learned	930	122	990	315	173	45	911	1035
correct	473	44	226	123	39	4	397	360
precision	51%	36%	22%	39%	22%	8.8%	43%	34%
prec.top 20	90%	60%	75%	85%	70%	20%	85%	75%

Table 1: Evaluation of Ontopopulis for Portuguese

	person	weapon
seed terms	56	22
learned	578	900
correct	408	123
precision	71%	14%
prec.top 20	95%	60%

Table 2: Evaluation of Ontopopulis for Spanish

the linear pattern learning ”per se” and second, the overall performance of NEXUS in terms of precision and recall.

We used Ontopopulis to learn several semantic classes. For each semantic class we manually filtered out the wrong terms before adding the list to the NEXUS resources. Note that in our experiments we limited the manual intervention to deleting while no adding or correction was allowed. In such a way we wanted to obtain a resource whose elements are all learned automatically.

We learned a dictionary of words and multiwords referring to people (e.g.“enviado especial”). This dictionary is used intensively by the person recognition grammar. Moreover, it is the longest dictionary exploited by NEXUS and its manual creation would be quite time consuming. On this point, the application of Ontopopulis was very important. Another semantic class to learn was the class *weapons*, which NEXUS uses to detect the means by which violent acts were committed.

Additionally, we learned several other semantic classes to be used in the process of event classification, namely *politician*, *vehicle*, *watercraft*, *edged weapon*, *crime* and *building*. Event classification is performed by a set of over 30 event category definitions, which are composed of boolean operators over keywords. Category definition designing is usually a time consuming manual process which requires both domain knowledge and language competence. We tried to partially automatize this process by converting category definitions into more abstract boolean expressions over semantic classes, which we could learn by our semantic category learning algorithms. We do not report here about the performance of the overall event classification, but we show accuracy figures for the learning of these semantic classes.

For each semantic class, we provided a set of seed templates and run Ontopopulis. As training data we used two unannotated corpora - 3,4 million titles of news articles for Portuguese and 5,7 million news titles for Spanish. The results for Portuguese and Spanish are shown in Table 1 and Table 2, respectively.

For each semantic category we report the number of seed terms, the number of the new terms learned by the system, the number of correct learned terms, the overall precision and the precision in the top 20 ranked terms. The accuracy in the top 20 seems to be quite high for most of the categories with exception of *weapons* and its subclass *edged weapons*. The overall precision is lower, since the system threshold was set very low in order to increase the recall and add more resources for the event extraction system. This was safe since we manually clean the Ontopopulis output in a last step. However, the fact that the accuracy is relatively high in the top 20 shows that the system properly orders the learned terms by putting the most reliable ones on the top.

Another positive outcome of the application of Ontopopulis was that we increased the size of the term lists between 2 and 13 times for most of the categories, after manually validating the system output.

5.2 Evaluation of linear pattern learning

We run linear pattern learning in order to obtain linear patterns for extraction of several domain-specific semantc roles: DEAD, WOUNDED, KIDNAPPED, ARRESTED and PERPETRATOR. As an example, for the *dead* role one of the Portuguese patterns the system learned was

	dead	wounded	kidnapped	arrested	perpetrator
seed patterns	12	7	31	10	38
learned	382	104	178	78	113
correct	54	11	24	28	19
precision	14%	11%	13%	36%	17%

Table 3: Evaluation of pattern learning for Portuguese

	dead	injured	arrested
seed terms	22	25	15
learned	108	10	15
correct	30	5	9
precision	28%	50%	60%

Table 4: Evaluation of pattern learning for Spanish

	DEAD	WOUNDED	KIDNAPPED	ARRESTED
baseline Portuguese	0.62	0.53	0.54	0.29
target Portuguese	0.69	0.51	0.67	0.47
baseline Spanish	0.12	0	0	0.125
target Spanish	0.46	0	0	0.125

Table 5: Evaluation of extraction of different roles in terms of F1-measure

“assassinato do [PERSON]”.

The experiments we report about here consisted of one learning iteration only. After that we manually filtered out unappropriate patterns. The results in Table 3 and Table 4 show the performance of the pattern learning algorithm for Portuguese and Spanish language, respectively - before the manual validation[3].

5.3 Evaluation of NEXUS

Test data were gathered by downloading EMM article clusters during 30 consecutive days in April 2009. The final test corpus was selected from these clusters as a sample of 100, which report about security and disaster-related topics.

On this corpus, we ran a baseline version of the system for both languages, namely the one based on seed linear patterns and seed dictionaries of persons and weapons. We also ran a target version in which we added to the seed resources the cleaned output of Ontopopulis for the classes *person* and *weapons* and the output of the pattern learning algorithm. We denote the baseline and target system with BL and TG, respectively.

Table 5 shows a comparative evaluation of the two baseline and target event extraction systems for Portuguese and Spanish.

In particular, we measured Precision (P), Recall (R) and F-measure for each role. We only show F-measure figures for a more compact comparison. Moreover, test data were slightly sparse, as some of the roles were not instantiated in text - namely RELEASED and PERPETRATOR - due to the relatively small corpus size. Therefore we do not report about them in the final evaluation.

Evaluation was done separately for each role, and data were collected cluster by cluster. Namely, for each cluster of articles we record if it contains a reference to the filler of a specific role; then we record if the system detected any filler whatsoever for that role, and finally, we record a correct detection if the returned role filler description equals at least one of the descriptions occurring in the cluster.

The comparative evaluation of the Portuguese baseline and target systems clearly shows that the target system performs better. On average, the F-measure improved by 0.09 in the target system. The maximal improvement was for the category ARRESTED - the F-measure improved from 0.286 for the baseline system to 0.47 for the target one. The average recall improvement was found to be 12%. The best improvement of the recall was for the role KIDNAPPED - from 60% to 80%. Moreover, the improvement in the recall was not at the cost of reduced precision, as on average the precision still improved by about 1%. Even if these results can be improved further, they demonstrate that machine learning algorithms bring improvement in the overall performance of the event extraction system. Data are less impressive for Spanish, and more sparse. Nonetheless, an even larger improvement in terms of F-measure could be recorded for the DEAD role.

[3] Results are only partial for Spanish due to data sparseness of the training corpus.

6 Conclusions

We presented a multilingual methodology for adapting an existing event extraction system to Portuguese and Spanish languages. The approach relies on weakly supervised learning of domain-specific lexicons, and requires minimal amount of domain and linguistic knowledge.

In our experimental settings, we only performed one learning stage, with no fine-tuning. Therefore, system performance in absolute terms was not excellent. Nonetheless, we believe that figures on the improved performance of the learned systems are encouraging, so that we plan to pursue in optimizing the development process. Moreover, the approach seems to be portable in the same way over semantic domains. One possible research direction would be then to test the methodology on adapting the event extaction system to new application domains.

The live event extraction system for Portuguese is publicly accessible at `http://press.jrc.it/geo?type=event&format=html&language=pt`. For the Spanish version change the value of the language attribute to es.

References

Aone, C. and M. Santacruz. 2000. Rees: A large-scale relation and event extraction system. In *Proceedings of ANLP 2000, 6^{th} Applied Natural Language Processing Conference*, Seattle, Washington, USA.

Appelt, D. 1999. Introduction to information extraction technology. Tutorial held at IJCAI-99.

Bell, A. 1991. *The Language of News Media.* Blackwell.

Erjavec, Tomaz. 2004. Multext - east morphosyntactic specifications. http://nl.ijs.si/ME/V3/msd/html.

Grishman, R., Huttunen S. and R. Yangarber. 2002. Real-time event extraction for infectious disease outbreaks. In *Proceedings of Human Language Technology Conference*, San Diego, USA.

Grishman, R., Huttunen S. and R. Yangarber. 2003. Information extraction for enhanced access to disease outbreak reports. *Journal of Biomedical Informatics*, 35(4).

Ji, H. and R. Grishman. 2008. Refining event extraction through unsupervised cross-document inference. In *Proceedings of 46^{th} Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, Ohio, USA.

King, G. and W. Lowe. 2003. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57:617–642.

Naughton, M., Kushmerick N. and J. Carthy. 2006. Event extraction from heterogeneous news sources. In *Proceedings of the AAAI 2006 workshop on Event Extraction and Synthesis*, Menlo Park, California, USA.

Piskorski, J. 2007. Express extraction pattern recognition engine and specification suite. In *Proceedings of the International Workshop Finite-State Methods and Natural language Processing*, Potsdam, Germany.

Piskorski, J. 2008. Corleone core linguistic entity online extraction. *Technical Report EUR 23393 EN.*

Piskorski, Jakub, Hristo Tanev, and Pinar Oezden Wennerberg. 2007. Extracting violent events from on-line news for ontology population. In *BIS*, pages 287–300.

Riloff, E. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence.*

Samuel, E., E. Ranchhod, H. Freire, and J. Baptista. 1995. A system of electronic dictionaries of portuguese. *Lingvisticae Investigationes*, XIX:2.

Shi, Lian, J. Sun, and H. Che. 2007. Populating crab ontology using context-profile based approaches. *Knowledge Science, Engineering and Management, LNCS.*

Steinberger, R., B. Pouliquen, and E. van der Goot. 2009. An introduction to the europe media monitor family of applications. In *Information Access in a Multilingual World - Proceedings of the SIGIR 2009 Workshop*, Boston, USA.

Tanev, Hristo and B. Magnini. 2006. Weakly supervised approaches for ontology population. In *Proceedings of the European Chapter of the Association of Computational Linguistics*, Trento, Italy.

Wagner, E., Liu J. Birnbaum L. Forbus K. and J. Baker. 2006. Using explicit semantic models to track situations across news articles. In *Proceedings of the AAAI 2006 workshop on Event Extraction and Synthesis*, Menlo Park, California, USA.

Yangarber, R., Jokipii L., A. Rauramo, and S. Huttunen. 2005. Extracting information about outbreaks of infectious epidemics. In *Proceedings of the HLT-EMNLP 2005*, Vancouver, Canada.