Evaluation of a Complex Information Extraction Application in Specific Domain

Romaric Besançon, Olivier Ferret, Ludovic Jean-Louis

CEA, LIST, Vision and and Content Engineering Laboratory
CEA Saclay Nano-INNOV, 91191 Gif-sur-Yvette, France
firstname.lastname@cea.fr

Abstract

Operational intelligence applications in specific domains are developed using numerous natural language processing technologies and tools. A challenge for this integration is to take into account the limitations of each of these technologies in the global evaluation of the application. We present in this article a complex intelligence application for the gathering of information from the Web about recent seismic events. We present the different components needed for the development of such system, including Information Extraction, Filtering and Clustering, and the technologies behind each component. We also propose an independent evaluation of each component and an insight of their influence in the overall performance of the system.

Keywords: Information Extraction, Evaluation, Filtering

1. Introduction

Information Extraction (IE) deals with the identification of structured information from unstructured text. It covers various tasks from Named Entity Recognition (NER) up to scenario template construction (Cunningham, 2005). For these different tasks, numerous approaches have been proposed and evaluated in general frameworks and evaluation campaigns such as MUC (*Message Understanding conference*) (Grishman and Sundheim, 1996), ACE (*Automatic Content Extraction*) or, more recently, TAC (*Text Analysis Conference*). These campaigns give the benchmarks needed to evaluate different tasks of IE, but when it comes to operational applications, intelligence tools for information extraction in specific domains are developed using numerous natural language processing technologies, that cover different tasks of IE, along with other modules for information filtering, retrieval or clustering.

Such systems can be evaluated directly using a end-user evaluation. However, this kind of evaluation is costly and cannot be used repeatedly during the development and tuning of the system. Furthermore, end-user evaluation also relies on the evaluation of ergonomic aspects through the quality of the user interface and not only on the quality of the results produced by the system. End-users evaluation should then be used for final evaluation of the system more than during the development, in order to improve the quality of the IE tools.

On another hand, the automatic evaluation of such complex frameworks is difficult because of the diversity of the integrated modules. A challenge for this evaluation is to take into account the limitations of each of these technologies in the global evaluation of the application. We present in this article an evaluation of an intelligence application in specialized domain that combines the independent evaluation of each component of the application, comparing various strategies for each component, and an error analysis that gives an insight on the influence of each component on the overall performance of the system. We present in section 2 the general architecture of the application, whose aim is to gather information from the Web about recent seismic events. Then, in section 3, we present the different components needed for the development of such system, for each component, the different methods tested and their evaluation. Finally, in section 4, we present an evaluation of the influence of each component on the global process and their contribution to the overall perceived quality of the system.

2. Presentation of the Information Extraction application

The Information Extraction application we evaluate in this article is designed to help analysts for the surveillance of seismic events. In this domain, the detection of new events is generally performed using signals from seismic and hydro-acoustic stations, treated with specialized analysis tools. The analysts also gather information from the Web to corroborate and complement the interpretation of the signals. The purpose of the application is to assist these analysts in linking the seismic information from the seismometers to information published on the Web, by identifying specific entities in the texts (locations, dates, magnitudes) and to structure these entities into event templates in order to present them to the user.

In the domain of seismic event surveillance, other works have been done to enrich the direct detection from sensors using texts (Sakaki et al., 2010; Earle et al., 2010) but these studies use more particularly Twitter as a source and rely on the information stream, treating temporal and spatial characterization of the tweets, and using the number of tweets rather than a sophisticated analysis of the content: in this case, Twitter itself is seen as another kind of sensor. In our approach, we use a deeper linguistic analysis of the texts, that is less adapted to a real-time detection of seismic events but is designed to provide a complement of information (the confirmation of a detection, how the people describes the event, the damages caused, etc.).

In this perspective, the objective of the application is to recognize, in texts extracted from the Web, the mention of a seismic event and to automatically identity relevant asso-

		Date Article	Titre	DATE	TIME	LOCATION	MAGNITUDE	DAMAGES	
	1	2009-02-03	Small earthquake rattles northern New J...	2009-02-03	06:10:00 UTC	northern New Jersey	3	no injuries or damages	
	1	2009-01-30	No Reports of Damage From 4.5-Magnitude...	2009-01-30	05:25:00 local	Seattle	4.5	no immediate reports of damage	
	1	2009-02-03	Earthquake of 4.2 magnitude shakes sout...	2009-02-02	07:09:00 local	southern Mexico	4.2	no casualties	
	1	2009-01-29	Magnitude-4.2 quake shakes isles off Ca...	2009-01-29	12:41:00 local	California	4.2	no apparent damages or injuries	
	3	2009-01-29	3.6 magnitude earthquake hits Northwest...	2009-01-28	01:30:00 local	Northwest Trinidad	3.6	no structural damage	
	1	2009-01-12	4.6 magnitude earthquake hits New Zeala...	2009-01-12	04:50:00 local	south of Nelson	4.6	no casualties or damages	
	3	2009-01-23	Strong earthquakes in Indian and Pacifi...	2009-01-23	09:00:00 local	north of Australia	6.2	no damage	
	2	2009-01-31	Friday quake revealed cracks in Wash. a...	2009-01-30	05:25:00 local	Washington	3		
	2	2009-01-29	2.5 Magnitude earthquake felt in North ...	2009-01-29	04:11:00 local	North Charleston	2.5		

Figure 1: Output of the application: synthetic presentation of the analyzed content in a dashboard

ciated information. The different modules involved in this application are the following:

- a collecting tool, that gathers texts from the chosen sources of information and extracts content-bearing text from their original format;
- a linguistic tool, that performs the linguistic analysis of the extracted text, and in particular the named entity recognition for the specific domain entities;
- an event identifier, that recognizes event mentions and link them with the relevant entities. This treatment is performed in two steps: first, a *segmentation* of the text separates the parts that are related to different events; second, a *slot filling* step chooses the interesting entities to attach to the event in the main event segment;
- a filtering tool, to filter out the non-relevant texts. This filtering uses two criteria: the first one is the detection of an event by the previous module, the second one uses a statistical classifier;
- a clustering tool, gathering the texts relative to the same event, in order to provide the user with a more synthetic view of the information.

Finally, the results of the information extraction process are presented in a synthetic dashboard where each line contains the different entities associated with an event, as presented in Figure 1. The application works on both English and French documents, but the evaluation presented in this paper only reports results on the French reference corpus.

3. Methods and Evaluation

All the evaluations presented in this paper have been performed on a corpus of French news articles concerning seismic events that have been collected between April and September 2008, on the French *Agence France Presse* (AFP) newswire (one third of the corpus) and from Google News (two thirds). The total corpus contains 501 relevant texts mentioning at least one seismic event. Other non-relevant texts have also been used to train the filtering tool.

3.1. Textual Content Extraction

The documents used for an IE application can come from different sources, such as newswires or news published on the Web. In the first case, the documents are generally well formatted and the content easy to extract (for instance, the news collected for this application from the AFP newswire are formatted using the XML format NewsML[1] from the IPTC). In the second case, we need to extract the interesting textual content from the HTML page. Otherwise, the surrounding text can add noise to the IE process (for instance, another event can be cited in the headline of a different article that is linked in the page).

Some works use machine learning techniques (for instance (Cai et al., 2003) uses the visual appearance of a Web page) that are dependent of the sites from which the pages are taken. We are in a context where we do not want to restricted ourselves to a given number of sites. Other approaches have been proposed, in particular in the context of the CLEANEVAL evaluation campaign on Web page cleaning (Baroni et al., 2008), but some strategies that achieve good performance in terms of coverage, such as *N-cleaner* (Evert, 2008), do not guarantee the readability of the result, which is fine when you want to use the extracted texts as a corpus to build language models, but is a problem in our application where the extracted text is presented to the end-users. An approach such as boilerpipe (Kohlschütter et al., 2010) combines shallow text features and text density features to identify the textual content and proposes a readable output.

We tested, in our application, two simple strategies for textual content extraction from Web pages:

- a first strategy, called *density-cleaner*, uses a text dump of the HTML page (using Lynx) and a measure of text density changes in the page, to spot the text borders. More precisely, we search for the strongest density changes in a similar way as (Hearst, 1997) for topic segmentation: we use lines of text as units and a sliding window on these units. The first most important increase of density indicates the beginning of the informative content part, the next most important decrease of density indicates the end of the informative zone. We also use additional indicators such as the presence of the title (when available as metadata) and

[1] http//www.newsml.org

the paragraph structure in order to adjust the limits to form a readable text;

- a second strategy, called *html-cleaner*, uses the HTML structure of the page to spot text markers in the page (such as <*br*> or <*p*> tags) and to go up to the closest common parent tag to get the text block. Using the HTML structure avoids the problem of detection of the right limit of the text which often arises with the first method. This strategy was inspired by the tool *Readability* (Arc90, 2009). We also use the presence of the title when available.

We tested these two methods on a corpus of 50 Web pages, which content has been annotated using the same format as in the CLEANEVAL evaluation campaign (Baroni et al., 2008), and used the scoring tool from this campaign, and compared them with the results obtained by N-cleaner and boilerpipe. The results are presented in Table 1. HTML-based cleaner gives better global results, comparable with boilerpipe with a simpler model. We also note that with our method, results a less regular: if the algorithm fails at spotting the textual content, either a wrong part of the page is returned, giving no interesting information, or the whole page is returned, giving too much noise. On a corpus of 500 documents, we estimated such cases at 1% for each kind of error.

	Precision	Recall	F-measure
density-cleaner	56.5%	90.4%	69.5%
html-cleaner	96.5%	94.4%	95.4%
boilerpipe	92.5%	97.1%	92.7%
N-cleaner	64.9%	83.7%	73.1%

Table 1: Evaluation of the results of textual content extraction from Web pages

3.2. Named Entity Recognition

Named Entity Recognition (NER) is performed using the linguistic analysis tool LIMA (Besançon et al., 2010). In LIMA, the NER works using hand-written pattern-based rules. These rules rely on the characterization of specific linguistic units used as triggers and on the form of the local context around the triggers, and on additional gazetteers. The rules can also be associated with specific actions that allow to perform operations on the recognized entity, such as normalization operations (for instance, in order to normalize relative dates such as "Monday", knowing the date of the document). The entities of interest in seismic domain were defined with the analysts of the domain and are presented in Table 2.

A first evaluation has been performed on a corpus of 50 texts that have been completely annotated for named entities, with an average F-measure score of 84%. A second evaluation has been performed on 501 texts partially annotated for named entities, with only the annotation of the entities that are associated with the main event. For this second evaluation, only the recall is computed (precision is meaningless since all entities are not in the reference). The results are presented in Table 3.

Entity type	Explanation of the entity
EVENT_TYPE	type of the event (earthquake, tsunami ...)
LOCATION	location of the event: the location can be a precise place (city) or a more global place (country)
DATE	date of the event
TIME	time of the event
MAGNITUDE	magnitude
DAMAGES	damages caused by the event
GEO_COORD	geographical coordinates of the event (longitude/latitude)

Table 2: List of the specific entities of interest to characterize a seismic event

Even if the results are not as good as state-of-the-art results for more standard named entities, the results are correct and, more particularly, on the corpus of 501 documents, the recall rate is generally high, which is necessary for the next steps of the IE process. The worst results are obtained for the DAMAGES entities (63%), which are more difficult because of a greater variability in expressions (the reference annotations contain simple phrases such as "*1,000 houses damaged*" and more complex expressions including complete sentences such as "*two junior high school buildings respectively located in Sumalata and Tolinggula sub districts were destroyed*").

3.3. Event Identification

After the linguistic analysis of the text, we have the information about the specific entities. The event identification step must find which of these entities are related to the main event of the text: for the analysts, the useful information is only the information relative to the most recent event. The event identification process is performed in two steps:

- we first segment the text into events: we focus on the extraction of information related to one particular event and the news articles often mention several events of the same type, for comparison purposes (the impact of a recent earthquake is compared with a more important earthquake that occurred previously in the same region). We therefore want to isolate parts of the text in which a single event is mentioned, and to do so, we focus on temporal information to segment the text in parts that are temporally homogeneous (each event is generally unique in a given time interval described in the text);

- in the segment referring to the main event (i.e. the most recent one), we choose from the entities which ones are related to the event in order to fill the slots of the event template.

3.3.1. Segmentation

For event segmentation, we want to distinguish segments relative to the main event, to a different event or to anything else. We use temporal information, with the hypothesis that parts of the text sharing the same temporal frame will deal with the same event.

	complete_50			partial_500
entity type	Precision	Recall	F-measure	Recall
EVENT_TYPE	93.9%	93.0%	93.4%	97.4%
LOCATION	90.5%	66.5%	76.6%	84.4%
DATE	88.2%	86.3%	87.2%	98.7%
TIME	82.6%	86.5%	84.5%	96.5%
MAGNITUDE	93.8%	83.3%	88.2%	94.0%
DAMAGES	83.5%	63.9%	72.4%	62.7%
GEO_COORD	100.0%	66.7%	80.0%	86.7%
All	89.8%	77.4%	83.2%	72.9%

Table 3: Evaluation of named entity recognition on 50 texts with complete annotation and 500 texts with partial annotation

Two methods have been tested. The first one is based on a heuristic temporal segmentation based on the presence and values of dates, with the following principles: dates with different values[2] correspond to different segments and the limit between two different segments is chosen between two different dates based on the structure of the text in sentences and paragraphs, along with the presence of other entities that are characteristic of the domain. The second method tested is based on a machine learning model, using temporal cues as features (verb tenses, presence of dates and temporal expressions) and a Conditional Random Field (CRF) model to take into account the sequence of the temporal information. This machine learning method aims at classifying each sentence of the text into one of the following classes: "*main event*" "*secondary event*", "*background*". This segmentation method is described in more details in (Jean-Louis et al., 2010). The results of the two methods are presented in Table 4, on a subset of the corpus containing 140 documents, manually annotated into segments. Most documents contain at least two events. The results show that the CRF model outperforms the heuristic method.

	heuristic		CRF	
event type	Recall	Precision	Recall	Precision
main event	82.8%	64.7%	98.7%	87.4%
sec. event	23.5%	43.4%	52.7%	95.8%
background	16.9%	21.7%	69.3%	92.7%

Table 4: Results of event-based segmentation

3.3.2. Slot Filling

For the slot filling part, we tested several strategies. The first method, called *Position* is a simple heuristic consisting in taking, for each entity needed to characterize the event, the first entity of the required type in the segment of text associated with the main event. A second set of strategies is composed of graph-based techniques based on the graph of relations between entities. This entity graph is built using statistic classifiers trained to indicate the presence or absence of a relation between two entities. A selection of the entities associated with the event mention in the this graph is performed using the connections in the graph and their weights. More specifically, we tested a selection method simply based on the weight of the relations (*Confidence*), a method based on the PageRank algorithm (*PageRank*) and a hybrid method combining the different selections according to the type of the entity (*Hybrid*). These strategies are described in more details in (Jean-Louis et al., 2011). The results, presented in Table 5, show that the simplistic heuristic already gives good results but can be improved using the more sophisticated techniques, the hybrid method giving the best results.

	Recall	Precision	F-measure
Position	73.4%	73.1%	73.2%
Confidence	74.9%	74.2%	74.5%
PageRank	72.4%	71.7%	72.0%
Hybrid	**77.6%**	**76.9%**	**77.2%**

Table 5: Evaluation of the association of entities to events for the 501 annotated documents

3.4. Filtering

Document filtering allows to keep only relevant news in the synthetic dashboard. For the documents collected from the Web, a first pre-filtering step can be integrated if we use a search engine to acquire the documents by defining a specific query relative to the domain. But it is generally difficult to have a non-ambiguous query. Furthermore, this pre-filtering step cannot be applied on documents collected blindly from a newswire, except if we decide to index all documents using a local search engine, which can be costly. A second straightforward criterion for this filtering is the effective discovery of an event by the previous module. But in practice, this filtering is not sufficient: we measured that, after this first filtering, 60% of documents are still non relevant. Among non-relevant documents, some use domain-related terms figuratively ("*political earthquake*"); others refer to an actual event but only anecdotally. A second filtering step has been integrated, using a statistical classifier trained on an annotated corpus made of texts selected after the first filtering step[3]. Following the existing work in the domain of text classification (Lewis et al., 2004), we

[2] Values of relative dates are normalized.

[3] The statistic filtering is used after the information extraction step instead of directly after the crawling in order to have a comparable corpus whatever the source of the documents is (Web or newswire), *i.e.* whatever the documents are pre-filtered by a query or not.

	Precision	Recall	F-measure	Accuracy
original training corpus				
words / presence / threshold = 0	94.3%	72.5%	82.0%	88.7%
words / presence / threshold optimal	80.6%	**91.2%**	85.6%	89.1%
words / tf.idf / threshold = 0	97.1%	72.5%	83.0%	89.5%
words / tf.idf / threshold optimal	93.5%	79.1%	85.7%	90.7%
lemmas / presence / threshold = 0	**98.5%**	71.1%	82.6%	89.5%
lemmas / presence / threshold optimal	84.5%	91.1%	**87.7%**	91.0%
lemmas / tf.idf / threshold = 0	**98.5%**	72.2%	83.3%	89.8%
lemmas / tf.idf / threshold optimal	93.6%	81.1%	86.9%	**91.4%**
words / presence / threshold = 0	94.3%	72.5%	82.0%	88.7%
words / tf.idf / threshold = 0	97.1%	72.5%	83.0%	89.5%
lemmas / presence / threshold = 0	**98.5%**	71.1%	82.6%	89.5%
lemmas / tf.idf / threshold = 0	**98.5%**	72.2%	83.3%	89.8%
modified training corpus				
lemmas / presence / threshold = 0	97.1%	75.6%	85.0%	90.6%
lemmas / presence / threshold optimal	85.9%	87.8%	86.8%	90.6%

Table 6: Evaluation of different parameters for the SVM filtering classifier

used a standard SVM (Support Vector Machine) classifier (Joachims, 1998) trained with 501 relevant documents and 711 non-relevant documents. A distinct test corpus of 91 relevant documents and 166 non-relevant documents was also built. As an implementation, we used the SVM^{Light} tool. A study of various parameters has been performed in order to optimize the performance of the classifier: in particular, the following parameters have been tested:

- the units used to represent the text: either the inflected forms of the words or their normalized forms (lemmas), obtained after the linguistic analysis of the texts (in this case, the specific entities have also been considered as units);
- the weighting of these units: we tested a simple binary weight indicating the presence/absence of the units in the text and a frequency based *tf.idf* weighting scheme (combining the frequency of the term in the document and the inverse document frequency of the term in the training set);
- an optimization of the decision threshold of the SVM: previous studies have shown that the default threshold of the model (= 0) is not always the optimal solution (Shanahan and Roma, 2003). Thus, we have set the threshold from the training corpus by optimizing a given criterion on a varying scale of thresholds (we used the F-measure as this criterion);
- a modification of the training corpus: the optimal filtering threshold is used to separate the training corpus into positive and negative examples and re-learn the model. By integrating more heterogeneity into the examples, this method allows the models to be more general (and should thus increase recall).

Table 6 shows that the best balance between precision and recall is obtained using lemmas with a binary weighting and an optimized threshold. The option *lemmas-tf.idf* gives better accuracy, but in this case, the recall has been considered too low by the end-users. The modification of the training corpus did not achieve better results.

3.5. Clustering

On the basis of the information gathered by the IE module, we can group the documents relative to the same event to provide the user with a more synthetic view of the information. More precisely, we used the dates and locations of the events as core information (considering the other entities or their evaluation may vary in different documents for the same event). These core entities were integrated using various methods:

- *evt(...)*: we simply use the equality of dates and locations of the event, supposing that other related information (magnitude, damages) may change in time;
- *section(...)*: in order to increase the coverage of the clustering, this method is designed to correct possible mistakes in the event identification step. The values used for the clustering are all entity values from the segment of the main event and a majority vote is used on all common entities;
- *document(...):* following the same idea, we extend the majority vote to all common entities in the document to correct the errors from the segmentation step.

Furthermore, these entities are subject of an additional normalization. The relative dates are normalized according to the publication date of the documents, such that every date in the text has an associated form month/day/year. The locations are also normalized using a geographical database built from the Geonames[4] database and containing links of spatial inclusion. Location names are often ambiguous (for instance, there are more than 70 places called *Paris* in Geonames) and we first disambiguate the location names using an algorithm inspired from (Pouliquen et al., 2006)

[4]http://www.geonames.org/

	Precision	Recall	F-measure	NMI
evt(DATE,LOC)	**87.8%**	23.8%	37.4%	0.84
section(DATE,LOC)	59.8%	43.8%	50.5%	0.80
document(DATE,LOC)	39.0%	53.8%	45.2%	0.79
evt(DATE,COUNTRY)	86.8%	56.7%	**68.6%**	**0.90**
section(DATE,COUNTRY)	42.8%	**62.7%**	50.1%	0.81
document(DATE,COUNTRY)	38.6%	55.1%	45.4%	0.79
Markov Clustering	50.3%	33.7%	40.4%	0.72
DBSCAN	67.5%	17.4%	27.6%	0.82
KMeans	53.5%	8.9%	15.3%	0.78

Table 7: Evaluation of different strategies for the event-based clustering

that uses, on one hand, probabilities associated with location types and importance (*Paris* is more probably the capital of France than a smaller place in another region of the world) and, on the other hand, measures of spatial consistence on the global text, based on the geographical distance of places spotted in the text (*Paris* may refer to a city in Texas if all the other places in the text are associated with Texas or the United States). The previous clustering techniques may then use either directly the location name (LOC) or the country associated with this location (COUNTRY), to have a more flexible matching between locations. We compare these clustering approaches to standard clustering algorithms, including the standard K-Means, Markov Clustering (van Dongen, 2000) and DBSCAN (Ester et al., 1996). The last two have the interest of not requiring an *a priori* fixed number of clusters. We tested these algorithms using either the titles only or the full texts, with inflected forms or lemmas. The best results were obtained on full texts with lemmas and are the only ones presented in the results.

The evaluation has been performed on the 501 news of our corpus, manually clustered into 142 different clusters (with 59 clusters containing more than 1 document), using Precision, Recall and F-measure on document pairs (a pair of documents is considered as correct is the two documents are part of the same cluster in reference and in test) and Normalized Mutual Information (Strehl and Ghosh, 2003). In Table 7, we see that the simple clustering gives good precision but poor recall. With the location name normalization, the improvement of coverage is obtained without an important loss in precision (only one point) and F-measure is then largely improved. These results also show that the quality of the results obtained by previous steps is sufficient to obtain good clustering results with a simple heuristic, better than with a standard clustering algorithm on full text.

4. Influence of Components in Global Performance

The output of the application is the synthesized information presented to the end-user in the dashboard. The overall quality of the application will then be assessed from this dashboard. All the components of the overall system may influence this quality. Actually, the most important component in this respect is the filtering. Indeed, its impact on the final IE result is quite straightforward and important, since each non-relevant document occupies a line in the table, contributing to a general impression of mistakes in the page. As we indicated in section 3.4, we noted that without the statistical filtering, there was around 60% documents kept for the IE process that were not relevant. We indeed measured that for the same IE method, the overall quality of the dashboard doubled when adding the statistical filtering step.

The quality of the textual context extraction may also influence the information extraction process. Table 8 presents the results of the information extraction on the 50 documents used for cleaning evaluation, using the different cleaning techniques presented in section 3.1 and comparing them with the results obtained with no textual content extraction (using only the dump produced by the Lynx text browser) and the results obtained using the reference of manually cleaned pages (*ref-clean*). Results are a bit inferior to the results presented in the rest of the paper because the IE reference was produced on a particular page cleaning result (which may induce a bias). However, we see that the quality of the textual content extraction does have an influence of the results: the results of the different techniques are mostly comparable, even if the html-cleaner performs a bit better, but all these results are better than the one obtained on the whole page, which confirms the need to clean the page. The improvement observed when using the text from the reference cleaning is relatively limited, which show that the cleaning strategies used are sufficient to get most of the relevant information.

	Precision	Recall
density-cleaner	63.2%	53.6%
html-cleaner	63.2%	55.4%
boilerpipe	62.8%	55.1%
lynx	56.3%	50.0%
ref-clean	64.8%	56.8%

Table 8: Evaluation of the influence of the quality of the textual content extraction on the IE process

As far as the IE process itself is concerned, the quality of the segmentation has an impact on the slot-filling step. Table 9 presents the results of the slot-filling (using the simple heuristic), with the two segmentation methods (heuristic and CRF-based), compared to the manual reference segmentation. These results show that the segmentation based

on machine learning performs a bit better, but the two results are comparable.

	Recall	Precision
without segmentation	66.6%	63.5%
heuristic	71.0%	68.6%
CRF	71.7%	68.8%
reference segmentation	87.5%	86.3%

Table 9: Impact of the segmentation on the slot-filling task

A complementary error analysis has been performed in order to determine the contribution of the various modules of the IE system. In this analysis, errors correspond to incorrect entities for an event, *i.e.* a line of the dashboard. They are characterized as follows:

- at least one entity of the reference for a given type has been identified in the main event segment, *i.e.* the error comes from the slot filling module;
- at least one entity of the reference of a given type has been identified in another segment, *i.e.* the error comes from the segmentation module;
- none of the reference entities has been identified in the text, *i.e.* the error comes from the named entity recognition module.

The distribution of the errors on these three types is given in Table 10, cumulated for all entity types, for the two segmentation strategies considered and the *Position* slot-filling strategy. This analysis shows that while the segmentation errors are divided by two with the CRF segmentation method, the global percentage of correct entities is quite similar for the two systems, mostly because the segmentation errors are partially turned into slot filling errors in the main event segment. Table 11 gives the error analysis with the CRF-based segmentation and the *Hybrid* strategy presented in section 3.3.2. It demonstrates that these errors can be partially corrected using advanced slot-filling techniques but that there is still room for improvement. By the way, it confirms that the slot filling task is still the most difficult task of an IE system[5].

	heuristic	CRF
correct entities	71.6%	72.2%
slot filling errors	18.6%	20.6%
segmentation errors	6.2%	3.8%
NER errors	3.6%	3.4%

Table 10: Analysis of the different error types in the event template construction

	Hybrid
correct entities	75.1%
slot filling errors	21.2%
segmentation errors	0.8%
NER errors	2.8%

Table 11: Analysis of the different error types in the event template construction

[5]The segmentation errors in this table are less important than in previous table because this error analysis has been performed with an different (improved) CRF segmenter.

5. Conclusion

We have presented in this paper an evaluation of an Information Extraction application in specialized domain, designed for the identification of events in news articles. This application deals with some difficult problems of IE: for instance, the texts in this domain usually mention several events, which adds ambiguity in the event identification and makes the event template construction more delicate. Moreover, this application, in order to be operational, integrates several components of natural language processing, text classification and clustering. The evaluation of the global application is difficult since each component has its own limitations and weaknesses. The evaluation we performed mainly relies on the independent quantitative evaluation of each component of the application but also includes a global error analysis to understand the part of the errors in the final output of the system that are due to each component. Such dual evaluation is particularly useful during the development of the application.

The next steps in the development of the application are focused on the information extraction part, and more specifically on the slot-filling task, where most of the errors are now occurring. The first perspective is the use of proximity and linguistic criteria in the event template construction (including syntactic relations). The document clustering is also a component where there is room for improvement, for instance by integrating the structured information extracted from the event in a more generic clustering environment (using a standard clustering model). From a more general perspective, we are interested in the possibility to overcome the specificities of the target domain for the IE application and to be able to build a more generic model that could be adapted to different domains using only limited supervision.

6. Acknowledgments

The research leading to these results has received funding from the European Union's Seventh Framework Program (FP7/2007-2013) under grant agreement n^o SEC-GA-2009-242352.

7. References

Arc90. 2009. Readability - An Arc90 Lab Experiment. http://lab.arc90.com/experiments/readability/.

Marco Baroni, Francis Chantree, Adam Kilgarriff, and Serge Sharoff. 2008. Cleaneval: a Competition for Cleaning Web Pages. In *Proceedings of LREC'08*, Marrakech, Morocco, may.

Romaric Besançon, Gaël de Chalendar, Olivier Ferret, Faiza Gara, Olivier Mesnard, Meriama Laïb, and Nasredine Semmar. 2010. LIMA: A Multilingual Framework

for Linguistic Analysis and Linguistic Resources Development and Evaluation. In *Proceedings of LREC'10*, Valletta, Malta, may.

Deng Cai, Shipeng Yu, Ji rong Wen, and Wei ying Ma. 2003. Extracting content structure for web pages based on visual representation. In *Proceedings of the 5th Asia Pacific Web Conference*, pages 406–417.

Hamish Cunningham, 2005. *Encyclopedia of Language and Linguistics*, chapter Information extraction, automatic. Elsevier.

Paul Earle, Michelle Guy, Richard Buckmaster, Chris Ostrum, Scott Horvath, and Amy Vaughan. 2010. OMG Earthquake! Can Twitter Improve Earthquake Response? *Seismological Research Letters*, 81(2):246–251.

Martin Ester, Hans-Peter Kriegel, Jrg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press.

Stefan Evert. 2008. A Lightweight and Efficient Tool for Cleaning Web Pages. In *Proceedings of LREC'08*, Marrakech, Morocco, may. European Language Resources Association (ELRA).

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471, Morristown, NJ, USA.

Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Ludovic Jean-Louis, Romaric Besanon, and Olivier Ferret. 2010. Using Temporal Cues for Segmenting Texts into Events. In Hrafn Loftsson, Eirkur Rgnvaldsson, and Sigrn Helgadttir, editors, *7^{th} International Conference on Natural Language Processing (IceTAL 2010)*, volume 6233 of *Lecture Notes in Computer Science*, pages 150–161. Springer Berlin / Heidelberg.

Ludovic Jean-Louis, Romaric Besanon, and Olivier Ferret. 2011. Text Segmentation and Graph-based Method for Template Filling in Information Extraction. In *5^{th} International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 723–731, Chiang Mai, Thailand.

Thorsten Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *10th European Conference on Machine Learning (ECML'98)*, pages 137–142, Berlin. Springer.

Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining (WSDM'10)*, pages 441–450, New York, NY, USA. ACM.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397.

Bruno Pouliquen, Marco Kimler, Ralf Steinberger, Camelia Ignat, Tamara Oellinger, Flavio Fluart, Wajdi Zaghouani, Anna Widiger, Ann charlotte Forslund, and Clive Best. 2006. Geocoding multilingual texts: Recognition, disambiguation and visualisation. In *Proceedings of LREC 2006*, Głnes, Italie.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of WWW'10*, pages 851–860. ACM.

James G. Shanahan and Norbert Roma. 2003. Improving SVM Text Classification Performance through Threshold Adjustment. In *Proceedings of ECML'2003*, pages 361–372.

Alexander Strehl and Joydeep Ghosh. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, March.

Stijn van Dongen. 2000. *Graph Clustering by Flow Simulation*. Ph.D. thesis, University of Utrecht, May.