Text Segmentation and Graph-based Method for Template Filling in Information Extraction

Ludovic Jean-Louis, Romaric Besançon, Olivier Ferret

CEA, LIST, Vision and Content Engineering Laboratory

Fontenay-aux-Roses, F-92265 France

{ludovic.jean-louis,romaric.besancon,olivier.ferret}@cea.fr

Abstract

In event-based Information Extraction systems, a major task is the automated filling from unstructured texts of a template gathering information related to a particular event. Such template filling may be a hard task when the information is scattered throughout the text and mixed with similar pieces of information relative to a different event. We propose in this paper a two-step approach for template filling: first, an event-based segmentation is performed to select the parts of the text related to the target event; then, a graph-based method is applied to choose the most relevant entities in these parts for characterizing the event. An evaluation of this model based on an annotated corpus for earthquake events shows that we achieve a 77% F1-measure for the template-filling task.

1 Introduction

Information Extraction (IE) is a process that aims at extracting pieces of information from texts. Following the paradigm defined in the Message Understanding Conferences (MUC) (Grishman and Sundheim, 1996), IE systems focus on extracting structured information concerning events to fill predefined *templates*. These templates make it possible to highlight the information that is specific to a type of events and to discard pieces of information that are not relevant in this respect. Figure 1 gives an example of the filling of a template by information extracted from a news article.

Common issues addressed by IE systems for filling a template include identifying named entities, finding relations between these entities, resolving entity coreference, gathering scattered information, etc. (Turmo et al., 2006).

Currently, there is no standard approach for filling templates. However, most IE systems have

Figure 1: Example of template filling

been relying on a sentence-oriented approach: first, domain-specific patterns or classifiers are used to process sentences separately; then, ad-hoc strategies for merging these local results into templates are applied. Even if such approach has been used widely, it does not take into account two important problems: *(i)* events can be described in more than one sentence; *(ii)* patterns/classifiers mainly capture binary relations among entities while events are not limited to binary relations.

An illustration of the first problem is given by Figure 1, where information relative to the event *EV1* is expressed beyond the sentence scope. This problem raises more generally the question of defining event-related spans of text or, in other words, determining whether a sentence refers to an event, and eventually the type of this event. In this article, we tackle this issue through the means of discourse segmentation. More specifically, we propose segmenting texts according to the events they refer to. Our objective is to narrow the span of text to explore in order to link a named entity to an event mention. As time is an important feature for discriminating events, we chose to perform this segmentation by relying on temporal cues.

Concerning the second problem, we can observe in Figure 1 that most of the sentences comprise an event mention with more than 2 related entities: the first sentence involves 3 entities while

the second one involves 4 entities. Similarly to (McDonald et al., 2005), we refer to such relations as *complex relation*s, namely any *n*-ary relation among *n* typed entities. In this context, each event can be seen as a complex relation where the arity of the relation *n* is equal to the number of entity types that should be filled in the template (*n=5* in the previous example). Several methods were proposed for extracting complex relations such as graph-based methods (McDonald et al., 2005; Wick et al., 2006) or inference-based methods (Goertzel et al., 2006). In this article, we tackle the complex relation extraction by proposing a graph-based method. We start by building an entity graph based on the result of text segmentation; then we use several domain-independent strategies for the reconstruction of the complex relation.

The remainder of this article is organized as follows: the next section discusses related work while Section 3 presents a general overview of our approach. Sections 4 and 5 detail the methods used for the two steps: event segmentation and template filling. Finally, Section 6 gives the results of the evaluation of each step.

2 Motivation and Related Work

Template filling is a central task for IE systems and has been the object of numerous studies. For instance, in the context of the MUC (Message Understanding Conferences) and ACE (Automatic Content Extraction) (Doddington et al., 2004) evaluation campaigns, one of the objectives assigned to the systems was to fill predefined templates with a static/fixed structure. Although this is the most widespread approach, a work such as (Chambers and Jurafsky, 2011) adopts a different view and proposes an unsupervised approach for filling templates without prior knowledge about their structure: they rely on clustering techniques for learning the structure of templates and on syntactic patterns for extracting their fillers.

A wide range of IE systems, particularly those based on learning approaches, have been relying on the idea that an event is often described within a single sentence, which leads to the hypothesis that pieces of information across sentences are less important. This idea is called the *single sentence assumption* by Stevenson (2006), who reported that only 60% of the facts mentioned in three MUC corpora (MUC 4-6-7) could be identified following this hypothesis. More recently, Ji et al. (2010) showed that around 40% of relations among entities require using cross-sentence inference techniques for their extraction.

Few approaches have been proposed for information extraction at a discourse level. Among them, (Gu and Cercone, 2006) and (Patwardhan and Riloff, 2007) are the closest to ours. (Gu and Cercone, 2006) is a segment-based HMM (Hidden Markov Model) approach. It relies on a first HMM model for identifying text units (sentences) that are relevant for the extraction of template fillers and on another HMM to extract the fillers from the retrieved sentences. Similarly, Patwardhan and Riloff (2007) proposed first to identify relevant sentences by using a self-trained SVM (Support Vector Machine) and then, to apply extraction patterns (primary and secondary patterns) to find the template fillers.

One of the first successful approach for the extraction of *n*-ary relations came from the biomedical community (McDonald et al., 2005) and was later applied to the domain of corporate management successions (Afzal, 2009). Other works tackled the complex relation problem in the context of *database record extraction*. They proposed to focus on the compatibility of a set of entities rather than on the compatibility of pairs of entities, which led them to take into account intersentential relations between entities (Wick et al., 2006; Mansuri and Sarawagi, 2006; Feng et al., 2007).

3 Overview

Event extraction as presented in this article takes place in a wider context of technology watch in which users are mainly interested in the most recent events. In this context, our goal is to synthesize from news articles the information about such recent events into a dashboard. However, news articles often refer to several comparable events, generally for pointing out the similarities and differences between a recent event and past events. In our specific application, we are not interested in the past events and we consider them as a source of noise for extracting information about the main event of a news article. We made the assumption, as in (Feng et al., 2007), that one document is associated with one record (event in our case). We adopted a two-step strategy to tackle this problem:

- segmenting texts into events: events might

be described over a single sentence. Therefore, we need to segment texts according to the events they refer to. These segments are frequently discontinuous as the structure of news articles is often dominated by moves between the main event and past similar events;

- filling event templates from relations: since event segments go beyond the sentence level, they are even more likely to contain complex relations than sentences. Therefore, we have to verify which entities mentioned in these segments are eligible to be part of the complex relation.

4 Segmenting Texts into Events

The goal of our segmentation of texts is to delimit segment units in relation with a target event. In previous work such as (Gu and Cercone, 2006; Patwardhan and Riloff, 2007), the methods for identifying such segments relied on fully lexicalized models that were learned using word surface forms. (Naughton, 2007) adopted a more generic approach by exploiting text structure. Our proposal is based on the idea that using temporal cues can help discriminate events, in particular similar events. In the example of Figure 2 for instance, two kinds of temporal cues can be used for this task: date values and verb tenses.

$^{\{MAIN\}}$ An earthquake measuring 5.6 on the Richter scale hit Jayapura, Papua, shortly after midnight on **Sunday**.
$^{\{SUB\}}$ Previously on **Saturday** the agency recorded a magnitude 5.6 earthquake had hit Melonguane in North Sulawesi.
$^{\{OUT\}}$ Indonesia, sits on a vulnerable quake-hit zone called the Pacific Ring of Fire.

Figure 2: Example of text segmentation: {MAIN}=Main event, {SUB}=Secondary event, {OUT}=Background

Our segmentation approach is based on an event-oriented representation of texts: a text is viewed as a sequence of sentences in which each sentence is characterized by the presence or the absence of an event. As in previous work, we have made the hypothesis that one sentence is linked to one event[1]. Hence, we tackle the segmentation

---
[1]This hypothesis is not verified for all texts but can be considered as a reasonable simplification in the context of our study.

task as a classification problem where each sentence of a text must be associated with an event type.

We classify sentences according to the following three categories. **Main event**: all sentences referring to the main event of the text; **Secondary event**: all sentences containing data that are related to an event different from the *Main Event*; **Background**: all sentences that belong neither to the *Main Event* nor a *Secondary Event*. An example for each category is given in Figure 2.

Our intuition is that for segmenting texts, the most interesting criteria rely not only on the nature of sentences but also on their linking at a discourse level, with the idea that categories of events don't follow one another in an arbitrary way. For instance, in the example of Figure 2, the shift from one event to another is associated with the change of verb tense *preterit/past perfect*. Our focus compared to previous segmentation approaches is to capture the dependencies between the shifts of temporal frames and the shifts of events.

For this purpose, we trained a linear Conditional Random Field (CRF) model (Lafferty et al., 2001) using the following temporal cues as features. *verb tenses*: a binary feature is associated with each possible verb tense (feature is 1 if at least one verb of the sentence has the considered tense); *presence of dates*: if a sentence contains a date, it is likely to refer to an event different from the previous sentence (except for the first occurrence of a date); *temporal expressions*: this feature accounts for the presence of temporal expressions in a sentence, such as *"over the past two weeks, in recent years"*, often related to generalities. The dependencies between successive event types are taken into account by the linear structure of our CRF model. More details about this segmentation model can be found in (Jean-Louis et al., 2010).

5 Filling Event Templates from Relations

For the filling of event templates, we propose a graph-based approach relying on the paradigm of complex relation extraction. Its first step (*graph construction*) detects relations between entity pairs in the same sentence to build an entity graph. The second step (*template filling*) applies generic strategies for selecting the most relevant entities associated with the template by exploiting the entity graph. These two steps are described in more details in the following sections.

5.1 Graph Construction

The entity graph we build in this first step characterizes at the document level the presence/absence of semantic relations between each pair of entities. For building such a graph, we propose to rely on the most relevant text segments in relation to the target event instead of considering the entire document. In our case, these segments are those built from the sentences classified as {MAIN} by the segmentation model presented in Section 4.

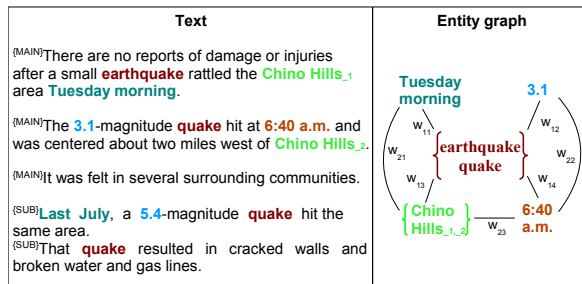| Text	Entity graph
{MAIN}There are no reports of damage or injuries after a small **earthquake** rattled the Chino Hills_1 area Tuesday morning. {MAIN}The 3.1-magnitude **quake** hit at 6:40 a.m. and was centered about two miles west of Chino Hills_2. {MAIN}It was felt in several surrounding communities. {SUB}Last July, a 5.4-magnitude **quake** hit the same area. {SUB}That **quake** resulted in cracked walls and broken water and gas lines.	Tuesday morning 3.1 w_{11} w_{12} w_{21} {earthquake quake} w_{22} w_{13} w_{14} Chino Hills_{1,2} w_{23} 6:40 a.m.

Figure 3: Example of entity graph

The entity graph is a weighted graph whose nodes are associated with named entity mentions while edges are associated with the relations between these mentions. The weight associated with each edge measures the confidence that a semantic relation exists between its two entities in a sentence. The graph is undirected as we mainly rely on relation confidence, a symmetric notion in our case, for filling templates. Figure 3 shows an example of such entity graph. Note that we assume that all entity mentions having the same value are equivalent (as the two mentions of *Chino Hills*) since they are located in the same event segment. Similarly, we consider all event mentions as equivalent (as *earthquake* in the first sentence and *quake* in the second sentence).

The presence of a relation between two entities in a sentence is classically determined by a statistical classifier. Following the standard approach of (McDonald et al., 2005) or (Liu et al., 2007), the weight of a relation is evaluated by the confidence score of this classifier and ranges in [0,1] in all the experiments of Section 6.3. In most previous works, such classifier mainly relies on a set of lexicalized features, without any syntactic feature (Afzal, 2009; Gu and Cercone, 2006; Wick et al., 2006). In (Liu et al., 2007), syntactic features are used in addition to lexicalized features. In contrast, our aim is to build a model that only makes use of syntactic features and does not rely on lexical information (such as inflected forms or lemma) in order to have a more generic model that can be easily adapted to another domain. For evaluating the contribution of lexicalized features compared to syntactic features, we trained different classifiers based on three distinct sets of features, detailed in Table 1:

Features description	FEAT-BASE	FEAT-LEX	FEAT-NOLEX
Entity type of E1 and E2	✓	✓	✓
POS of E1 and E2	✓	✓	✓
Words in E1 and E2	✓		
Word bigrams in E1 and E2	✓	✓	✓
Words between E1 and E2	✓	✓	
Word bigrams between E1 and E2	✓	✓	
POS between E1 and E2	✓	✓	✓
# words between E1 and E2	✓	✓	✓
POS bigrams between E1 and E2		✓	✓
# synt. relations between E1 and E2		✓	✓
Syntactic path between E1 and E2		✓	✓
Relative position and POS[2]		✓	✓
# entities between E1 and E2		✓	✓
# event mentions between E1 and E2		✓	✓
POS of two words after/before E1		✓	✓
POS of two words after/before E2		✓	✓

Table 1: Features for relation classification

- *FEAT-BASE*: same feature set as (Afzal, 2009), based on lexicalized features;

- *FEAT-LEX*: a feature set that contains lexicalized features, and syntactic features inspired by (Liu et al., 2007)[3];

- *FEAT-NOLEX*: same feature set as *FEAT-LEX*, but without the lexicalized features.

5.2 Template Filling

Template filling aims at selecting the best entities for the template slots. In our approach, this selection relies on the relations between entities in the entity graph. Note that we are trying to fill domain-specific templates that have a fixed number of slots though it is not mandatory that every slot gets a value. As every slot is associated with an entity type, we need to select (when it is possible) one entity value for each slot. This problem can be seen as the complex relation reconstruction task described in (Afzal, 2009; McDonald et al., 2005). We compared several approaches to tackle this issue:

---

[2]If either E1 or E2 is an event mention, indicate whether the other entity is after/before its POS.

[3]Some of their features are not relevant in our context since they are only applicable in the biomedical domain.

Heuristic is a simple but efficient approach that selects for each entity type the first entity mention occurring in the main event segment.

Confidence is an approach that selects, for each entity type, the entity connected to the event mention with the highest confidence weight.

PageRank is a link analysis based approach that relies on the PageRank algorithm (Brin and Page, 1998). The idea is to use the graph structure to rank entities according to their importance in the graph and to select, for each entity type, the entity mention with the highest PageRank score.

Vote is a voting-based approach exploiting the output of the *Confidence*, *PageRank* and *Heuristic* approaches: a majority vote is performed for each entity type and the entity mention with the highest number of votes (one vote by approach) is chosen.

Hybrid is an hybrid approach that applies for each entity type the best selection approach for this type. The main idea is to increase overall performance by allowing one entity type to be selected with one approach and another entity type to be selected by a different approach. For instance, the best selection approach for dates is *Confidence* whereas it is *Heuristic* for geographical coordinates.

Except for the first approach, the output is complemented by the use of the heuristic approach as back-off when no entity has been retrieved for a given entity type. Such a situation happens when a template filler is the only entity of a sentence and therefore, cannot be extracted by the binary relation classifier.

6 Evaluation

This section provides details concerning the experimental evaluation of our template-filling process. We present the corpus in Section 6.1 and the individual results for the different steps of our process in Sections 6.2, 6.3 and 6.4. We also evaluate the impact of the segmentation step on the final results in Section 6.5 and finally propose an analysis of errors in Section 6.6.

6.1 Corpus

The work presented in this paper was developed for an application dedicated to the surveillance of earthquake event mentions in news articles. The earthquake event template summarizes the main characteristics of a seismic event, namely its date, time, location, magnitude, geographical coordinates and its mention (earthquake, afterquake, etc.)[4]. An example of such template is provided in Figure 1, knowing that our target application is not interested in the secondary event *EV2*.

We carried out all the experiments on a corpus of 501 French news articles concerning earthquakes. These articles were collected between February 2008 and early September 2008 from a *Agence France Presse* (AFP) newswire (1/3 of the corpus) and from Google News (2/3 of the corpus). The corpus was manually annotated by domain analysts for filling the earthquake event template. The annotators identified a total of 2,775 entities divided into 6 entity types: event mention (18%), location (34%), date (17%), time (12%), magnitude (17%) and geo-coordinates (1%)[5].

Each document was preprocessed by the LIMA linguistic analyzer (Besançon et al., 2010), performing tokenization, sentence boundary detection, part-of-speech tagging, verb tense analysis, named entity recognition and dependency parsing.

6.2 Segmenting Text into Events

We used a subset of 140 articles from our corpus as training data for the CRF-based segmentation model. These articles were manually annotated into 1,659 segments according to the categories defined in Section 4: *Main event (70%), Secondary event (17%), Background (13%)*. Most of these articles contain a main event and at least one secondary event (short articles might not refer to a secondary event). Note that the *Secondary event* class includes without distinction all secondary events. The implementation was achieved using the CRF++[6] toolkit. We report in Table 2 results of our CRF model (*CrfSeg*) compared to a baseline (*ParaSeg*) in terms of F1-measure using a 5-fold cross-validation. The baseline *ParaSeg* is a paragraph-based heuristic that assigns *Main event* category to all the sentences in the first two paragraphs and considers others

---

[4]Casualties were not considered here because their correct identification would require a chunker.

[5]Several entities could be annotated for the same slot when variants or different levels of granularity were present: for locations, both a city and a country name for instance.

[6]http://crfpp.sourceforge.net

as secondary event[7]. Results in Table 2 show

Event type	ParaSeg (%)	CrfSeg (%)
Main event	11.14	92.71
Secondary event	21.7	67.91
Background	0	79.42

Table 2: Results of text segmentation into events (F1-Measure)

that our model obtains fairly good classification performance for all categories and is particularly suited for identifying the main event section. They also show the impact of taking temporal cues into account compared to relying on text structure only. Note that the poor results of our baseline partly come from its ignorance of the *Background* category. The *Relevant Sentence Classifier* of (Patwardhan and Riloff, 2007) has a goal similar to our segmentation with Recall|Precision|F1-Measure scores of 63%|46%|53% on terrorism documents and 72%|41%|52% on disease outbreak documents. We provide these figures as indicative results but not for direct comparison since their approach is different: they used a SVM classifier with lexicalized features and not temporal cues, classified sentences according to two classes (Irrelevant, Relevant) and performed their evaluation for English, on the MUC-4 terrorism and ProMed corpora.

6.3 Graph Construction

Our graph construction method relies on a binary relation classifier for assessing the existence of a semantic relation between two entities in a sentence. We experimented several types of statistical classifiers with the three sets of features (*FEAT-BASE, FEAT-LEX, FEAT-NOLEX*) presented in Section 5.1. A set of 44 articles from our corpus was used to annotate binary relations. Among the 5,000 binary relations in these articles, 969 were in-sentence relations. 43 relations were discarded because one of their entities was actually included in the span of a larger entity not recognized because of its type (such as organizations), the rest was used for training the classifiers: 690 in the *POSITIVE* class, in which the two entities of the candidate relation refer to the same earthquake event, and 236 in the *NEGATIVE* class, where the

two entities are associated with different earthquake events. The following sentence contains examples of both *POSITIVE* and *NEGATIVE* relations:

[POSITIVE]: The first *quake*, with a magnitude of *5.3*, struck at about 11.05am and was followed a few minutes later by a stronger quake with a 6.5 magnitude...

[NEGATIVE]: The first *quake*, with a magnitude of 5.3, struck at about 11.05am and was followed a few minutes later by a stronger quake with a *6.5* magnitude...

We tested three types of learning algorithms with these quite unbalanced training data by relying on the Mallet toolkit[8] for their implementation: Naive Bayes (*NB*), Maximum Entropy (*ME*), Decision Tree (*DT*). We report in Table 3 the results obtained for each feature set and algorithm in terms of recall (*R*), precision (*P*) and F1-Measure (*F*) using a 5-fold cross-validation. The results of a simple baseline that assigns the *POSITIVE* category to each relation are also given.

Feature set	Algo.	R(%)	P(%)	F(%)
FEAT-LEX	ME	96.30	95.92	96.10
FEAT-BASE	ME	91.22	96.09	93.57
FEAT-NOLEX	ME	91.66	94.99	93.26
FEAT-LEX	DT	89.01	96.45	92.55
FEAT-LEX	NB	93.44	90.69	92.02
FEAT-NOLEX	DT	91.17	88.74	89.83
FEAT-NOLEX	NB	89.58	89.23	89.37
FEAT-BASE	DT	84.35	94.70	89.16
FEAT-BASE	NB	86.73	87.86	87.27
Baseline	–	100.00	25.50	40.49

Table 3: Results for binary relation classifiers

Table 3 first shows the interest of using syntactic features as FEAT-LEX outperforms FEAT-BASE. Moreover, the non-lexicalized feature set FEAT-NOLEX obtains results equivalent to the lexicalized feature set FEAT-BASE. Concerning the learning algorithms, we observe the following hierarchy: ME > DT > NB. (Afzal, 2009) observes a different hierarchy, DT > ME > NB, but on a different corpus and a different language, which makes the comparison difficult. In terms of general performances, our results are in the same range as those reported in (Afzal, 2009), his best score being R=0.95%|P=0.87%|F=0.91% with a decision tree. Finally, we adopted the Maximum Entropy model trained with the FEAT-NOLEX feature set instead of FEAT-LEX. This choice is motivated by the fact that FEAT-NOLEX obtains

---

[7]We experimented other learning approaches such as HMM and Maximum Entropy models but we only provide results for the best approach, that is to say, CRF.

reasonable results without relying on strongly domain-dependent information such as lexicalized features.

6.4 Template Filling

As we mentioned in Section 5.2, our approach for template filling relies on the selection of relevant entities from the entity graph. Our idea is to compute for each entity a weight that quantifies its importance in the graph and consequently, makes it possible to rank the entities. We assume that the best ranked entities are more likely to be good fillers than others.

Before applying the entity selection strategies described in Section 5.2 to the entity graph, we apply a node merging step. The goal of this step is, on the one hand, to identify all the nodes that refer to the same entity value and remove duplicates and, on the other hand, to establish cross-sentence relations between entities. In our case, we used the node merging step for event mentions and date and location entity types: all dates having the same normalization and all locations having the same surface form are considered equivalent.

All annotated documents in our corpus were used for evaluating the different entity selection strategies. We report in Table 4 the results of template filling in terms of recall (*R*), precision (*P*) and F1-Measure (*F*), aggregated for all entity types.

Approach	**R(%)**	**P(%)**	**F(%)**
Hybrid	77.55	76.87	77.15
Vote	74.93	74.27	74.54
Confidence	74.89	74.16	74.47
Heuristic	73.40	73.06	73.17
PageRank	72.41	71.73	72.01

Table 4: Association of entities to events

These results confirm that the basic heuristic strategy is a powerful approach since it performs slightly better than the PageRank strategy. As the PageRank strategy only relies on the graph structure without considering the weight of the edges, its highest ranked entities are those that are highly connected regardless of the weight of the edges. As a consequence, if several non fillers entities are strongly linked, they get better scores than the others. Mihalcea (2004) proposed a weighted version of the PageRank algorithm that deals with this issue and should be tested in this context. Table 4

also shows that the best strategy is the *Hybrid* approach, which associates each entity type with a given selection approach: this method corrects the fact that an approach can perform well on a given entity type but poorly on another type.

6.5 Impact of Event Segmentation

In this section, we propose to evaluate the impact of our text segmentation procedure on the template filling task. Our text segmentation method focuses on identifying relevant text spans for the extraction. However, all documents do not mention several earthquake events and some only focus on a single event. In the latter case, our temporal segmentation might seem less relevant since all the sentences refer to the same event.

Our purpose in this section is to evaluate the impact of the segmentation on documents that mention a single event compared to those that mention multiple events. Our intuition is that the temporal segmentation should have a limited effect on single event documents and improve results on multiple event documents. In order to verify this hypothesis, we manually split the initial corpus into two sets according to the number of earthquake events they discuss. We obtained 227 multi-event documents (*M*) and 274 single-event documents (*S*). Finally, we applied each template-filling strategy on both (*M*) and (*S*) document sets, including the segmentation step or not. We report the results in Table 5 in terms of F1-Measure aggregated for all entity types.

| | **Without** | | **With** | |
	segmentation		**segmentation**	
Approach	S(%)	M(%)	S(%)	M(%)
Hybrid	79.20	73.61	78.34	75.61
Vote	77.67	68.68	76.89	71.81
Confidence	72.55	66.07	71.79	69.10
Heuristic	73.96	73.16	73.07	73.10
PageRank	70.92	59.72	70.67	65.32

Table 5: Impact of segmentation on single/multi-event texts (F1-Measure)

Concerning single-event documents, results in Table 5 show that the best performing strategies do not use segmentation though the global difference is not highly significant (+0.71% in average). At the opposite, strategies based on segmentation perform better on multi-event documents (+2.74% in average). Moreover, our best strategy (hybrid ap-

proach with segmentation) outperforms our baseline (heuristic without segmentation) on both document sets. Globally, these findings demonstrate that our temporal segmentation preserves results on single-event documents and improves results on multi-event documents.

6.6 Error Analysis

In order to have a more comprehensive view of the performance of our method for template filling, we performed an analysis of errors. The idea of this analysis is to identify the reason why a given entity filler is not found. In this context we identified three major types of errors:

- named entity recognition errors (*NE-err*): the entity is not identified by the linguistic pre-processing;

- event segmentation errors (*Seg-err*): the entity is identified by the linguistic pre-processing but its sentence is not tagged as part of a {MAIN} segment;

- template filling errors (*Fill-err*): the entity was identified in the correct event segment but was not selected as a template filler;

- *Correct*: the entity was correctly identified and selected as a template filler.

Figure 4 presents the percentage of each type of errors on all our evaluation corpus for two template filling strategies[9]: one without segmentation, the heuristic strategy *(Heuristic)*, and the other with segmentation, the hybrid strategy *(Hybrid)*. The
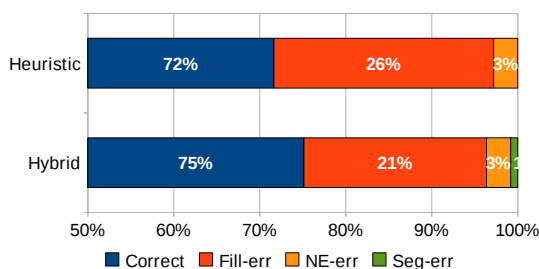

Figure 4: Classification of errors

graph shows that the baseline heuristic strategy achieves a high level of correctly identified entities (*72%*) but a significant level of template filling errors (*26%*). Our best strategy reduces this type of

---

[9]The percentages on the graph are rounded up, which explains why they do not sum to 100%.

errors while it increases the percentage of correct fillers. Moreover, it only induces a limited number of errors due to event segmentation (*1%*).

7 Conclusion

Most of IE approaches focus on sentence-based evidences for filling templates and rely on few discourse level information. In this article, we have presented an approach for template filling based on event segmentation and graph-based entity selection. Our event segmentation takes place at the discourse level and relies on temporal cues. It uses a CRF model to find the sentences that are most relevant for filling the event template. These sentences are then used to build an entity graph from which template filler entities are selected. We have proposed several strategies for selecting the entities – heuristic, confidence-based, PageRank-based – and various ways of combining these strategies: vote and hybrid approaches.

We have presented detailed results of our IE approach on a corpus of French news articles about earthquake events. Our experiments have shown that this approach improves the simple, but powerful heuristic in this field, that always selects the first entity found in a document for each type of fillers. These results have also shown that our approach is well suited for documents that mention several comparable events. Finally, our analysis of errors have demonstrated that there is still room for improvement since 21% of remaining errors are due to incorrect entity selection.

Concerning future work, our next experiments will be dedicated to the generalization of our template filling method to different contexts and more precisely, to other languages and domains. We have already obtained promising results by testing our event segmentation module on a set of English news articles in the seismic domain with only a limited effort of adaptation. For domain generalization, we are planning experiments in the financial domain.

References

Naveed Afzal. 2009. Complex Relations Extraction. In *Conference on Language & Technology 2009 (CLT09)*, Lahore, Pakistan.

Romaric Besançon, Gaël de Chalendar, Olivier Ferret, Faiza Gara, and Nasredine Semmar. 2010. LIMA: A Multilingual Framework for Linguistic Analysis and Linguistic Resources Development and Evaluation. In *7th Conference on Language Resources and Evaluation (LREC 2010)*, Valletta, Malta.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands. Elsevier Science Publishers B. V.

Nathanael Chambers and Dan Jurafsky. 2011. Template-Based Information Extraction without the Templates. In *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986, Portland, Oregon, USA.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation. In *4th Conference on Language Resources and Evaluation (LREC 2004)*, pages 837–840, Lisbon, Portugal.

Donghui Feng, Gully Burns, and Eduard Hovy. 2007. Extracting Data Records from Unstructured Biomedical Full Text. In *EMNLP-CoNLL'07*, pages 837–846, Prague, Czech Republic.

Ben Goertzel, Hugo Pinto, Ari Heljakka, Michael Ross, Cassio Pennachin, and Izabela Goertzel. 2006. Using Dependency Parsing and Probabilistic Inference to Extract Relationships between Genes, Proteins and Malignancies Implicit Among Multiple Biomedical Research Abstracts. In *HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 104–111, New York, USA.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: A Brief History. In *16th International Conference on Computational linguistics (COLING'96)*, pages 466–471, Copenhagen, Denmark.

Zhenmei Gu and Nick Cercone. 2006. Segment-based hidden Markov models for information extraction. In *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 481–488, Sydney, Australia.

Ludovic Jean-Louis, Romaric Besançon, and Olivier Ferret. 2010. Using temporal cues for segmenting texts into events. In Hrafn Loftsson, Eiríkur Rögnvaldsson, and Sigrún Helgadóttir, editors, *7th International Conference on Natural Language Processing (IceTAL 2010)*, volume 6233 of *Lecture Notes in Computer Science*, pages 150–161. Springer Berlin / Heidelberg.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2010. Overview of the TAC 2010 Knowledge Base Population Track. In *Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Eighteenth International Conference on Machine Learning (ICML'01)*, pages 282–289, San Francisco, CA, USA.

Yudong Liu, Zhongmin Shi, and Anoop Sarkar. 2007. Exploiting Rich Syntactic Information for Relationship Extraction from Biomedical Articles. In *NAACL-HLT'07, short paper session*, pages 97–100, Rochester, New York.

Imran R. Mansuri and Sunita Sarawagi. 2006. Integrating unstructured data into relational databases. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 29–40, Washington, DC, USA.

Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. 2005. Simple algorithms for complex relation extraction with applications to biomedical IE. In *ACL 2005*, pages 491–498, Ann Arbor, Michigan, USA.

Rada Mihalcea. 2004. Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. In *42st Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, Barcelona, Spain.

Martina Naughton. 2007. Exploiting Structure for Event Discovery Using the MDI Algorithm. In *45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 31–36, Prague, Czech Republic.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. In *EMNLP-CoNLL'07*, pages 717–727, Prague, Czech Republic.

Mark Stevenson. 2006. Fact distribution in Information Extraction. *Language Resources and Evaluation*, 40(2):183–201.

Jordi Turmo, Alicia Ageno, and Neus Català. 2006. Adaptive information extraction. *ACM Computer Surveys*, 38(2):1–47.

Michael Wick, Aron Culotta, and Andrew McCallum. 2006. Learning Field Compatibilities to Extract Database Records from Unstructured Text. In *EMNLP'06*, pages 603–611, Sydney, Australia.