ISSN 2007-9737

Complex Named Entities Extraction on the Web: Application to Social Events

Armel Fotsoh[1], Annig Le Parc Lacayrelle[2], Christian Sallaberry[2]

[1] RECITAL,
France

[2] Univ Pau & Pays Adour / E2S UPPA,
Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour,
France

armel@recital.ai, {annig.lacayrelle, christian.sallaberry}@univ-pau.fr

Abstract. In this paper, we focus on the extraction of social events in text from the web. We consider social events as complex Named Entities (NEs) i.e. NEs represented by a list of properties that can be simple values (text, number, etc.), "elementary" NEs and/or other complex NEs. Regarding the extraction of these complex NEs, our contribution focuses on the noisy context issue. Very few works in the state-of-the-art deal with this issue, and the few existing ones have limits in several contexts. We propose an original processing method based on supervised learning and patterns that makes it possible to focus property annotation on specific blocks of webpages. This process is generic and independent of the type of NE processed. We experimented and evaluated it with an example of complex NEs: social events. It appears that, in a noisy context, the results obtained with our approach considerably improve the standard process used in the state-of-the-art. The work was conducted with the objective of generalize it for other categories of complex NEs.

Keywords. Information extraction, complex named entities, social event.

1 Introduction

Recent developments in information technologies have made the web an important data source. Additionally, the number of contributors to this data source is increasing very rapidly and the published content is usually unstructured.

There are standards such as schema.org microdata [14] that have been defined to structure the information published on the web. However, these standards are only used very little in practice (less than 1% of websites use such standards).

Therefore, automatic processing of webpages for information extraction purposes is a difficult task. This is one reason for the rapid growth in the number of research works related to Information Extraction (IE) [4] in the textual content of webpages.

The information contained on the web generally refers to real-world objects such as people, places, points of interest (POI) or even events: in the state of the art, these objects correspond to Named Entities (NEs) [23].

However, a NE such as a social event can be described by a list of properties (e.g. its *title*, its *category*, its *location* and even its *performer*). In our approach, a NE, when described by several properties is called a *complex NE*.

Consider an excerpt of text from a webpage describing an event (see Figure 1). The extraction of different items of information on this event raises several issues. First of all, text phrases corresponding to properties of the event (*performer, category, location, etc.)* are drowned in the text without particular markups [19]. Furthermore, some properties can be expressed in

non-regular forms [8] (e.g., the event's title). Finally, some properties can be noisy.

This is the case of the event's performers, who are person (*Jeff Panacloc, Jean-Marc*). The text describing the event also contains other people (*Nicolas Nebot* and *Erwan Champigne*) who have nothing to do with the event performers. In the same way, the event date (*Saturday, October 14, 2017*) is described as noisy because, in the text, we have other dates as the ticketing opening date (*September 06, 2017*) and the date of another event with the same performer (*April 10, 2016*).

Ultimately, a complex NE's property is said to be noisy if the analysed text contains several candidate phrases that can match this targeted property. When this issue arises, the extraction context is referred as noisy. This paper will focus on this issue.

Fig. 1. Excerpt of text describing an event

We propose an approach in order to automatically extract complex NEs in webpages. The originality of this process lies in the fact that we have developed a processing method dedicated to the identification of noisy properties. This processing is based on supervised learning and patterns. It helps to isolate blocks of webpages potentially containing the properties of complex NEs. Only these blocks will then be analysed in order to identify the corresponding properties. We apply this generic approach to social events in order to evaluate the contribution of our proposal. The paper is presented as follows. We define the Named Entity concept in section 2, with a focus on the social event NEs. We look at works carried out on complex NE extraction in section 3.

In section 4, we present in detail our extraction process for social events, while section 5 focuses on the evaluation of the obtained extraction system. Section 6 concludes the paper and proposes new research perspectives.

2 Definitions

The notion of Named Entity (NE) emerged during the MUC (Message Understanding Conferences) [13] in the 1990s. Despite the various changes that it has undergone since then, this notion remains difficult to define, with several definitions being proposed [13, 5, 23]. Dupont [9] defines a NE as a linguistic unit of a referential nature which refers to people, organizations, places, dates, etc. In other words, a NE can therefore be defined as a linguistic unit (phrase), uniquely identifiable in a specific context and that refers to a real-world object. To process NEs, we organize them into two main categories: *elementary* and *complex*.

Elementary NEs are represented by a single phrase that refers to the object of the real world to which it corresponds. Therefore, an elementary NE can be a person (*Gustave Eiffel*), a place (*Liberty Island*), or a date (Saturday, October 14, 2017), when described by one property only.

A complex NE is a NE represented by a list of properties. Consider the NE corresponding to the movie "Star Wars". It can be described by 3 properties: its title (Star Wars), its director (Georges Lucas) and a set of categories (Action, Adventure). In this example, the title of the movie is a text, the director is an elementary NE of the person type and the category is a set of texts. The director, who is a person NE, can also be seen as a list of properties (name, date of birth and place of birth). In this case, he is no longer represented as an elementary NE but as a complex one.

In formal terms, let us consider a complex NE e represented by n properties ($n > 1$). We define it by: $e =< p_1, p_2, ..., p_n >$. The property p_i can be mono-valued or multi-valued (set). The value of property p_i can be an instance of a simple

type (text, number, etc.), an elementary NE, or a complex NE.

Before processing any complex NE, it is important to define models to represent them. In our application case, we deal with event NEs.

According to Zhang et al. [29], an event is "something that happens somewhere, at a given time and involves a certain number of actors". Two types of events are generally distinguished in the state of the art; these are facts and social events. Facts correspond to historical events, current topics, or even the episodes of a story. Social events refer to concerts, festivals, conferences, sport events, etc., to which a schedule and an audience are associated. For our application case, we deal with social events and propose a representation model for their description. This is given in Figure 2.

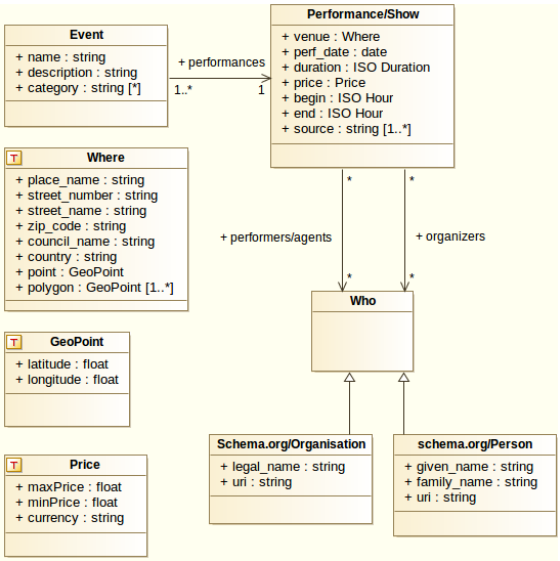

Fig. 2. Social event representation model

In our model, social events may occur on several occasions. Indeed, it is possible to attend the same concert of artist X on several dates and at different venues. For this reason, our model describes information about social events on two levels: (i) the event's thematic level, which serves to characterize it and to express its invariable properties such as the title or categories; (ii) the event's performance, which represents variable properties.

An event's performance is scheduled and is characterized mainly by a venue, a date and a set of performers (artists). The set of performers is attached to the performance and not to the event, because the performers may vary from one occasion to another. Indeed, the actors in a play staged by a touring theatre company may change. The same applies to the organizers. Other properties such as start and end times, duration or even ticket prices are taken into account.

3 Related Work

Several works have focused on Complex NE extraction in text from the web. Rae et al. [22], for example, address the extraction of POIs in web content, while Orlando et al. [21] and Foley et al. [10] focus on events. The proposed extraction processes are specific to the complex NE types analysed (POIs, events, etc.). However, the extraction process is usually carried out in two steps: (i) the text is processed in order to identify the different properties; (ii) and the identified properties are gathered to build complex NEs.

Regarding the first step, four main categories of approaches are generally used: pattern-based [2, 1, 26], knowledge-based [28, 25], learning-based [17, 24, 31] and hybrid [7, 3] which combines the previous three. We have seen in the example in Figure 1 that the properties of a complex NE are generally drowned in the text.

Therefore, to identify each of them, one or a combination of the four extraction approaches is implemented according to the specificity of the targeted property. When the targeted property follows very regular forms such as dates [26], it is usually the pattern-based approach that is implemented for their extraction. Next, when the targeted property can be described in a knowledge resource, knowledge-based approaches tend to be the most used [18]. When the targeted property cannot be characterized by regular forms (as is the case of the performers in Figure 1), a learning-based approach is generally implemented for its extraction [10].

Sometimes, some of the three first approaches are combined for property identification.

For instance, to reinforce the extraction of a property using a learning-based approach, resources can be used to introduce knowledge in the learning model training step. Rae et al. [22], for example, combine data from Wikipedia to train a learning model dedicated to the detection of POIs' locations in webpages.

Once extracted, the second step consists in gathering the identified properties in order to complete complex NE fields: here arises the noisy context issue which is only covered slightly in the literature. Indeed, most of the works related to complex NE extraction deal with not noisy corpora. As a matter of facts, for event extraction, these corpora include mostly short texts from social networks posts (Twitter, Facebook, etc.). However, Orlando et al. [21] have made a proposal for the extraction, in a noisy context, of event complex NEs such as *"Larry Page founded Google Inc. in 1998 with Offices in California"* in the print media. The extraction process is also divided into two stages: identification of properties (*"Larry Page"*, *"Google Inc"*, *"1998"*, *"California"*) and their aggregation.

Regarding the second step, they consider that the context may be noisy. After identifying the properties in text, they gather them using the Cartesian product to build candidate complex NEs. A relevance score is computed for each candidate by querying a search engine (Google) and projecting the properties of candidates on the pages yielded by the search engine. Only candidates with the highest relevance scores will be targeted as valid events. This process works properly in the context of Orlando et al. because the properties are strongly related to each other. Indeed, replacing for example *"Larry Page"* by *"Mark Zuckerberg"* in a candidate event and querying Google gives results that are far removed from the query: hence the low relevance score. This makes it possible to disqualify erroneous candidate entities.

However, in some cases, this process could lead to the extraction of the wrong complex NEs. Referring to the example in Figure 1, a candidate event built by taking the ticketing opening date as the event's opening date will obtain a good score with Orlando's approach. Most of the webpages that describe the event in Figure 1 also contain the ticketing opening date. Therefore, querying a search engine with this candidate event and the real one will lead to close scores, despite the fact that one of the two is erroneous.

4 Complex Named Entitiy Events Extraction Process

We have designed a processing chain dedicated to the extraction of social events in webpages (see figure 3). This chain takes into account the existence of noisy properties. Contrary to the standard approach which is based on two steps (property annotation and property aggregation), we introduce a new processing step that locates the relevant webpage blocks which may contain the property's phrases (block detection) before annotating them. This is interesting as the issue of noisy properties is solved before their annotation.

We will now detail each of the three modules of our processing chain.

4.1 Block Detection

The objective of this first module is to focus the annotation of properties on blocks likely to contain the right phrases. It takes as an input the social event representation model and the webpage to analyse. As an output of this first module, for each property described in the model, we have the webpage block in which it might be contained.

The observation of a sample of pages from the analysed corpus shows us that the position of the block containing properties can vary or not. For example, the event's title is always stacked in the header of the analysed webpage, for organic search reasons, while the position of category or date of events varies according to the analysed page and/or website. Therefore, we propose to use two strategies for the block detection:

— when the block containing a property is stacked in a specific position in the webpage, we propose a pattern-based approach;

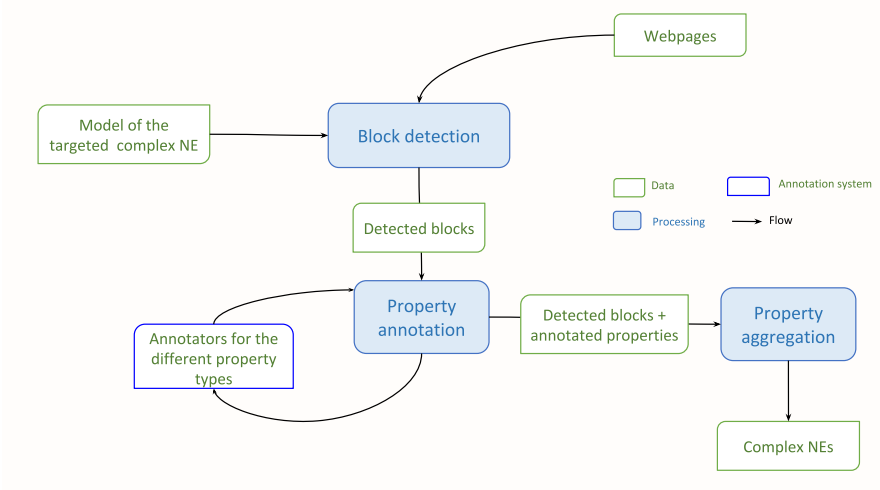

Fig. 3. Complex NE general extraction process

— when the position of the block containing a property varies, we propose a learning-based approach to locate it.

Thus, we implement a pattern-based strategy for the event's title identification. The corresponding pattern is given below:

$$< head > \quad Title\_Block \quad < /head > .$$

With regard to other properties such as category or date of events, we have carried out a supervised-learning-based strategy for their identification. We only analyse the textual content of webpages for this purpose. In the state of the art, there are several supervised learning-based algorithms that are known to be effective for text tagging. The two most commonly used ones are the Neural Networks algorithm [30] and the Conditional Random Fields (CRF) algorithm [16]. For our experimentation, we chose the CRF algorithm. This was due to the difficulty of compiling the necessary training set. In fact, Neural Networks require a large amount of data for training when the number of features to take into account is high [6]. In our context, the analysed texts are long (600 words average) and we wish to use the maximum number of features to detect blocks. CRF appears to be a good compromise between manual annotation efforts and efficient processing.

Several criteria are necessary for determining features to infer the learning model. We based ourselves on the work of Ollagnier et al. [20] for the choice of these criteria. The most relevant for our problem are the following:

— The word (token): this will generate features like the part of speech (POS), the lemma, the shape (upper case, lower case, capitalized), the position in text, etc;

— Sequence of words: this makes it possible to take into account the co-occurrences of words;

— Properties of the words in the same window: for each word in a window, joint features are constructed from those of other words. The number of words to take into account to the left and to the right of the analysed word are parameters of the algorithm;

— Substrings of characters constituting a word: in this case, the features applied to the words are also applied to their substrings (n-gram). The maximum size of substrings to consider is a parameter of the algorithm.

We use the CRF implementation of the Stanford NER library [1] to train our block detection model. The training set is composed of 360 webpages from 12 ticketing websites. This corresponds to about 218,000 words as input into the trainer.

The obtained blocks are illustrated in Figure 4. Indeed, blocks containing the effective properties of the event are delimited by continuous line rectangles. In this example, the identification of the *performance block* makes it possible to distinguish between the date of the described event (*Saturday, October 14, 2017*) and both the ticketing opening date (*September 06, 2017*) and that of the other event with the same performer (*April 10, 2016*) mentioned in the text. The same block also contains information about venue, performers and even price and hour.

This block detection module needs to be tuned each time a new corpus has to be processed: pattern-based and/or learning-based approaches might be adapted.

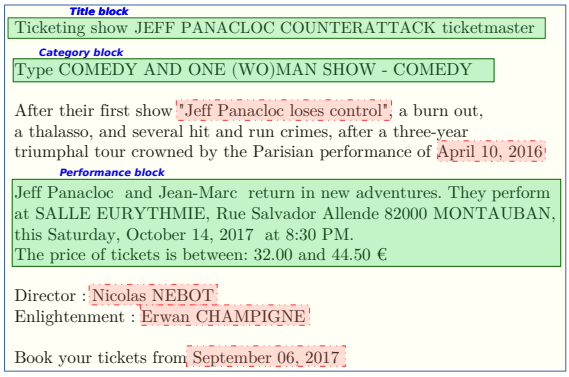

Fig. 4. Illustration of block detection

4.2 Property Annotation

This module is dedicated to the identification of the properties of complex NEs in the blocks detected by the first processing module. As an input, we have a set of annotators that can be state-of-the-art standards or customized ones. We also have the model describing each of the complex NE's properties and the webpage in which blocks have been detected.

For each property, the block containing it is selected first. Then an annotator in the input set is invoked according to the specificities of the property. Finally, the selected annotator is used to identify the targeted property in the relevant block.

To identify an event title in the corresponding block, we choose a supervised learning-based approach. This choice is due to the fact that this property does not follow any regular forms or cannot be expressed in a controlled vocabulary. Indeed, an event's title can be a single word (*Rihanna*) or a whole sentence (*Jeff Panacloc loses control*). We use the CRF algorithm to train our event title annotator, in the same way as for block detection. We re-use the same criteria for the generation of features, but with different parameters (window size, substring size, etc.).

The event category annotator is built using a knowledge-based approach. The main reason for this choice is that the vocabulary used to express a social event's categories is controlled. We analysed a set of online platforms dealing with this type of event to build a knowledge resource, including ticketing websites, tourist office sites and event directories. We represent this resource as a tree of concepts, each concept being characterized by a set of labels. The resulting tree has 4 hierarchical levels and contains 57 concepts for 610 labels. The resource is projected on the text of blocks to annotate the mentions of event categories.

The annotation of the event venues is based on the cascading combination of three annotators. The purpose of the cascading combination of these annotators is to detect an event venue in its most complete form whenever possible. Indeed, the first one in this combination is a pattern-based address annotator [11]. If it does not detect any place, an annotator based on a gazetteer of event venue names (built from DBPedia [2] and Google Maps [3] data) is then invoked. If no venue is detected by the first two annotators, an annotator using a city name lexicon can be used to identify the venue.

The invoked annotator for date identification is built using a pattern-based approach. This is because the dates follow very regular forms.

[1] https://nlp.stanford.edu/software/CRF-NER.shtml

[2] http://wiki.dbpedia.org/

[3] https://www.google.fr/maps

The same is true for price and hour annotators which are also built using patterns. For the performer annotation, we use two distinct annotators.

The first one exploits a thesaurus, built from DBpedia and YAGO [4] data, for the identification of performer names. The second one uses a supervised learning model that is trained with the CRF algorithm. To annotate performers in relevant blocks, the first annotator is invoked, then the second is also invoked to identify performers not referenced in the resource. The combination of these two annotators is intended to minimize silence during performer annotation.

The result is shown in Figure 5.

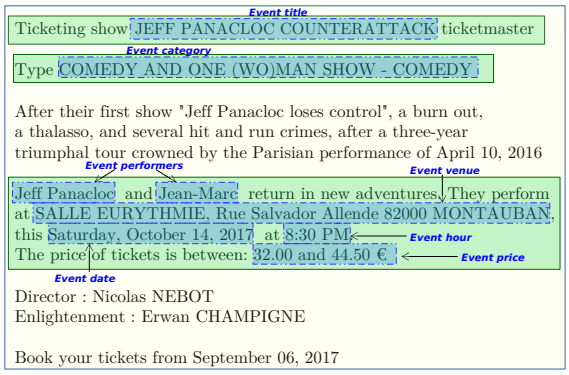

Fig. 5. Illustration of property annotation

4.3 Property Aggregation

Once the properties have been annotated in the corresponding blocks, the purpose of this last module is to aggregate them into the corresponding complex NEs. Two scenarios can be distinguished depending on whether the analysed webpage contains one or many complex NEs:

— If the webpage contains only one complex NE, aggregation simply associates each annotated property with the corresponding field according to the representation model of the complex NE;

— If the webpage contains many complex NEs, patterns or "dedicated" algorithms are more appropriate.

In our corpus, each analysed webpage describes a single event. Therefore, the aggregation process consists of parsing the webpage content and associating each of the annotated properties with the corresponding fields in the event's model (see Figure 2). Detecting the *performance block* makes it possible to group each of their properties (*date, venue, hour, etc.*) without ambiguity.

5 Evaluation

This section does not evaluate the existing complex NE annotators, nor customized ones. We simply implemented (i) the two-step state-of-the-art approach (i.e., annotation of properties first and then their aggregation) as well as (ii) our three-steps approch detailed before (Figure 3). Then, we analysed a corpus with both systems and compared the results (i.e., annotations obtained without and with a block detection module upstream). Indeed, we wanted to evaluate the ability of our annotation system to properly identify the right properties of event NEs, and more particularly to measure the contribution of the block detection module in a context where some properties can be noisy. This evaluation protocol is built based on the TREC [27] procedure.

5.1 Protocol

Scenarios

We have defined two evaluation scenarios:

— **Scenario 1**: the properties are annotated in the whole webpage text without any block detection (all the modules of the generic chain are instantiated except the block detection one);

— **Scenario 2**: the properties are annotated by analysing relevant webpage blocks. For this scenario, all the modules in our generic chain are instantiated, in particular the one dedicated to block detection.

[4] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/#c10444

Evaluated Properties

Here we will evaluate the extraction of five properties: *title, category, venue, date and performer*.

Corpus & Metrics

The evaluation corpus is composed of 150 webpages extracted from 12 ticketing websites. Each webpage describes a single event which may contain one or more performance(s). To compare our two scenarios, we use the three following evaluation metrics [12]: precision, recall and F_1-measure.

5.2 Results

We consider that the annotation of the properties is strongly linked to the detection of the blocks in which they are contained. As a result, we began our evaluation by measuring the quality of the block detection module. The category block and performance one (which contains the venue, the date and the performers) are characterized by the fact that their position varies in the text of the pages. Regarding the block containing the title, it corresponds to the *head* tag of the HTML page which is relatively simple and effective to detect. Our system always finds them in every page of this evaluation corpus.

Concerning the category and the performance blocks, the results of their detection on the evaluation corpus are given in Table 1. These results show that our block detection module is very accurate. So when a text block is tagged as containing a property, this is correct in more than 93% of the cases. Anyway, some blocks containing properties are not detected. This is mainly due to linguistic turns and the use of abbreviations.

Table 1. Evaluation of the block detection module

Blocks	Precision	Recall	F_1-measure
Category	93.25	82.17	87.36
Performance	94.58	83.16	88.50

Table 2 presents the results obtained from the experimentation of the two scenarios described above (standard approch compared with the one using the block detection module) for the property annotation. P, R and F_1 correspond to the results of scenario 1 and P', R' and F_1' to those of scenario 2. The gains obtained with the block detection module correspond to Δ_P, Δ_R, Δ_{F_1}.

The first observation that emerges from these results is the clear improvement in all the F_1-measures with the instantiation of the block detection module. This is because the block detection module is working properly (about 90% correctness): when it marks a webpage block as containing a given property, this is generally correct. However, this improvement impacts the recall of certain properties, including category, venue, date and performers. Indeed, if the block containing a property has not been detected, then the associated property will not be annotated, hence the decline in the recall. This problem may arise when the block containing the targeted property is detected using the learning-based strategy (category, venue, date and performer).

Regarding the annotation of the event title, the instantiation of the block detection module serves to increase both precision and recall. Indeed, analysing only the text of the page's header considerably reduces the risk of annotating wrong titles: this explains the increase in precision. In addition, analysing just a text fragment makes it possible to take advantage of learning performance to annotate information in short texts. As a result, this increases the title annotator recall. However, some event titles are not detected by our annotator even though they are present in the header of the page. These are usually very short titles (e.g. "P. Kass"), which are very uncommon in practice.

With regard to the category, the block detection module makes it possible to improve by about 50% the precision of the system compared with scenario 1. However, the category annotator sometimes does not detect labels in the selected blocks. This is mainly because the corresponding phrases are not listed in the resource. A lexical enrichment of the category resource is underway to reduce this silence.

For the venue, the precision obtained is about 95%, well above the 83% obtained on average by Jiang et al. [15], who experimented location

Table 2. Evaluation results

Property	Scenario 1			Scenario 2			Gains		
	P	R	F_1	P'	R'	F_1'	Δ_P	Δ_R	Δ_{F_1}
Title	75.26	48.67	59.11	92.57	87.63	90.03	+ 17.31	+ 38.96	+ 30.91
Category	42.10	92.05	57.77	92.45	76.15	83.51	+ 50.35	- 15.90	+ 25.73
Venue	51.02	95.28	66.46	94.25	81.16	87.22	+ 43.23	- 14.12	+ 20.76
Date	52.82	94.62	67.79	95.82	82.03	88.39	+ 43.00	- 12.59	+ 20.59
Performer	37.15	90.24	52.63	84.16	79.24	81.63	+ 47.01	- 11.00	+ 28.99

identification with several free Named Entities Recognition Systems (Stanford NER [5] and Spacy [6] in particular) on different corpora of texts. The cascade combination of three different annotators helps to limit the proportion of non-identified venues. Therefore, almost all the locations in the performance blocks are annotated by our system.

For dates, the block detection module helps to correctly isolate the relevant dates. This makes it possible to limit the proportion of erroneous dates, thus improving precision (43% gain). Moreover, as the dates follow very regular forms in our corpus, when the blocks containing them are detected, generally our annotator identifies them all.

As with the other properties, block detection helps to significantly improve the precision of performer annotation (47% gain). However, some erroneous annotations occur due to ambiguous cases. For example, suppose that the processed webpage describes a conference where "Jean Marc Verdier" is a speaker and is not referenced in the resource. Additionally, consider that the resource contains a performer whereby "Jean Marc" (the puppet of Jeff Panacloc [7], a French comedian) is one of the labels: the performer annotator will tag the phrase "Jean Marc" (the puppet) as the performer, although it has absolutely nothing to do with the conference.

6 Conclusion

We developed an approach for the extraction of social event complex NEs in webpages.

Several issues are taken into account in our proposal and the one we tackle in this paper is the noisy context problem. A complex NE's property is said to be noisy if the analysed text contains several candidate phrases that can match this targeted property. In related works, the extraction process is realized in two stages : the first one annotates the properties of complex NEs and the second one aggregates the annotated properties to build complex NEs. The noisy context issue is usually taken into account during this second stage. Our approach differs in the fact that we treat the noisy context before the annotation stage, starting by locating automatically the relevant webpage blocks which may contain the properties' phrases.

So, our processing chain consists of three main modules: (i) the first one detects webpage blocks that contain the properties of complex NEs; (ii) the second one analyses these blocks to annotate the properties; (iii) and the last module aggregates the annotated properties to build complex NEs. The introduction of the block detection module constitutes an improvement for complex NE extraction in a noisy context. Indeed, with the experimentation of our approach for the extraction of social events, we observe an average gain of 40% in precision, as well as an average gain of 25% in the F_1-measure compared to an approach without the block detection.

Our approach for social event complex NEs extraction in webpages is generic. It is important to note that the model of the targeted complex NE (e.g., social events) is one of the parameters of the block detection module (see Figure 3). However, as explained in the description of this module, each time a new corpus has to be processed,

[5] https://nlp.stanford.edu/ner/

[6] https://spacy.io/

[7] https://fr.wikipedia.org/wiki/Jeff_Panacloc

pattern-based and/or learning-based approaches might be tuned again.

This work was conducted with the objective of replicating the process for other categories of complex NEs. To evaluate the genericity of our approach, we experimented it on another category of complex NE. A first experiment on business entities showed results similar to those obtained for social events, especially for address and activity field properties annotation. Ongoing works now focus on new experiments in order to confirm the contribution of the block detection stage. Our ultimate goal is to define a generic extraction process suited to any category of complex NE.

References

1. **Alonso, O., Baeza-yates, R., Strötgen, J., & Gertz, M. (2011).** Temporal information retrieval: Challenges and opportunities. *In: 1st Temporal Web Analytics Workshop at WWW*, pp. 1–8.
2. **Blohm, S. (2011).** *Large-scale pattern-based information extraction from the world wide web*. KIT Scientific Publishing.
3. **Cellier, P., Charnois, T., Plantevit, M., Rigotti, C., Crémilleux, B., Gandrillon, O., Kléma, J., & Manguin, J.-L. (2015).** Sequential pattern mining for discovering gene interactions and their contextual information from biomedical texts. *Journal of biomedical semantics*, Vol. 6, No. 1, pp. 27.
4. **Chang, C.-H., Kayed, M., Girgis, M. R., & Shaalan, K. F. (2006).** A survey of web information extraction systems. *IEEE Trans. on Knowl. and Data Eng.*, Vol. 18, No. 10, pp. 1411–1428.
5. **Chinchor, N. & Robinson, P. (1997).** Muc-7 named entity task definition. *Proceedings of the 7th Conference on Message Understanding*, volume 29.
6. **Collobert, R. & Weston, J. (2008).** A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, ACM, New York, NY, USA, pp. 160–167.
7. **Dey, A., Paul, A., & Purkayastha, B. S. (2014).** Named entity recognition for nepali language: A semi hybrid approach. *International Journal of Engineering and Innovative Technology (IJEIT) Volume*, Vol. 3, pp. 21–25.
8. **Downey, D., Broadhead, M., & Etzioni, O. (2007).** Locating complex named entities in web text. *IJCAI*, volume 7, pp. 2733–2739.
9. **Dupont, Y. (2017).** *La structuration dans les entités nommées*. Ph.D. thesis, Paris 3.
10. **Foley, J., Bendersky, M., & Josifovski, V. (2015).** Learning to extract local events from the web. *38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, ACM, New York, NY, USA, pp. 423–432.
11. **Fotsoh, A. T., Sallaberry, C., Le Parc Lacayrelle, A., & Moal, T. (2016).** Extraction of business information on the web to supply a geolocated search service. *Proceedings of ALLDATA 2016, The Second International Conference on Big Data, Small Data, Linked Data and Open Data*, AllData '16, IARIA, Lisbon, Portugal, pp. 82–85.
12. **Gleverdon, C. W. (1962).** Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems.
13. **Grishman, R. & Sundheim, B. (1996).** Message understanding conference-6: A brief history. *Coling*, volume 96, pp. 466–471.
14. **Guha, R. V., Brickley, D., & Macbeth, S. (2016).** Schema.org: Evolution of structured data on the web. *Commun. ACM*, Vol. 59, No. 2, pp. 44–51.
15. **Jiang, R., Banchs, R. E., & Li, H. (2016).** Evaluating and combining named entity recognition systems. *Proceedings of the Sixth Named Entity Workshop, Joint with 54th Association for Computational Linguistics*, pp. 21–27.
16. **Klinger, R. & Tomanek, K. (2007).** Classical Probabilistic Models and Conditional Random Fields. Technical Report TR07-2-013, Department of Computer Science, Dortmund University of Technology.
17. **Le, N. T., Mallek, F., & Sadat, F. (2016).** Uqam-ntl: Named entity recognition in twitter messages. *WNUT 2016*, pp. 197.
18. **Lejeune, G., Brixtel, R., Doucet, A., & Lucas, N. (2015).** Multilingual event extraction for epidemic detection. *Artificial intelligence in medicine*, Vol. 65, No. 2, pp. 131–143.
19. **Nadeau, D. & Sekine, S. (2007).** A survey of named entity recognition and classification. *Lingvisticae Investigationes*, Vol. 30, No. 1, pp. 3–26.
20. **Ollagnier, A., Fournier, S., & Bellot, P. (2016).** Cascade de crfs et svm pour la détection

de références bibliographiques diffuses dans les articles scientifiques. *CORIA-CIFED*, pp. 139–152.

21. **Orlando, S., Pizzolon, F., & Tolomei, G. (2013).** Seed: A framework for extracting social events from press news. *22Nd International Conference on World Wide Web*, WWW '13 Companion, ACM, New York, NY, USA, pp. 1285–1294.
22. **Rae, A., Murdock, V., Popescu, A., & Bouchard, H. (2012).** Mining the web for points of interest. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, ACM, New York, NY, USA, pp. 711–720.
23. **Sekine, S., Sudo, K., & Nobata, C. (2002).** Extended named entity hierarchy. *LREC*.
24. **Shinyama, Y. & Sekine, S. (2004).** Named entity discovery using comparable news articles. *Proceedings of the 20th international conference on Computational Linguistics*, Association for Computational Linguistics, pp. 848.
25. **Soner, K., Ozgur, A., Orkunt, S., Samet, A., K., C. N., & N., A. F. (2012).** An ontology-based retrieval system using semantic indexing. volume 37, Oxford, UK, UK, pp. 294–305.
26. **Strötgen, J. & Gertz, M. (2010).** Heideltime: High quality rule-based extraction and normalization of temporal expressions. *Proceedings of the 5th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, pp. 321–324.
27. **Voorhees, E. M. & Harman, D. K. (2005).** *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press.
28. **Wang, W. & Stewart, K. (2015).** Spatiotemporal and semantic information extraction from web news reports about natural hazards. *Computers, environment and urban systems*, Vol. 50, pp. 30–40.
29. **Zhang, C., Zhou, G., Yuan, Q., Zhuang, H., Zheng, Y., Kaplan, L., Wang, S., & Han, J. (2016).** Geoburst: Real-time local event detection in geo-tagged tweet streams. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, ACM, pp. 513–522.
30. **Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998).** Forecasting with artificial neural networks: The state of the art. *International journal of forecasting*, Vol. 14, No. 1, pp. 35–62.
31. **Zhou, G. & Su, J. (2002).** Named entity recognition using an hmm-based chunk tagger. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 473–480.

Article received on 17/01/2019; accepted on 04/03/2019.
Corresponding author is Armel Fotsoh.