HAL open science

Extracting Event-related Information from a Corpus Regarding Soil Industrial Pollution

Chuanming Dong, Philippe Gambette, Catherine Dominguès

To cite this version:

Chuanming Dong, Philippe Gambette, Catherine Dominguès. Extracting Event-related Information from a Corpus Regarding Soil Industrial Pollution. KDIR 2021, Oct 2021, Setúbal, Portugal. pp.217-224, 10.5220/0010656700003064 . hal-03366097

HAL Id: hal-03366097

https://hal.science/hal-03366097v1

Submitted on 10 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Extracting Event-related Information from a Corpus Regarding Soil Industrial Pollution

Chuanming Dong[1,3][a], Philippe Gambette[2][b] and Catherine Dominguès[1][c]
[1]*LASTIG, Univ. Gustave Eiffel, ENSG, IGN, F-77420 Champs-sur-Marne, France*
[2]*LIGM, Univ. Gustave Eiffel, CNRS, ESIEE Paris, F-77454 Marne-la-Vallée, France*
[3]*ADEME, Agence de l'Environnement et de la Maîtrise de l'Énergie, F-49004, Angers, France*
chuanming.dong@ign.fr, philippe.gambette@univ-eiffel.fr, catherine.domingues@ign.fr

Keywords: Information Extraction, Deep Learning, Word Embedding, Semantic Annotation, Industrial Pollution.

Abstract: We study the extraction and reorganization of event-related information in texts regarding industrial pollution. The object is to build a memory of polluted sites that gathers the information about industrial events from various databases and corpora. An industrial event is described through several features as the event trigger, the industrial activity, the institution, the pollutant, etc. In order to efficiently collect information from a large corpus, it is necessary to automatize the information extraction process. To this end, we manually annotated a part of a corpus about soil industrial pollution, then we used it to train information extraction models with deep learning methods. The models we trained achieve 0.76 F-score on event feature extraction. We intend to improve the models and then use them on other text resources to enrich the polluted sites memory with extracted information about industrial events.

1 INTRODUCTION

Pollution is becoming one of the major concerns for French dwellers. The French Ministry of the Ecological Transition (MTES) is responsible for collecting and updating pollution data from industrial sites which are gathered in a certain number of databases, including BASOL, the database of (potentially) polluted sites; BASIAS, a historical inventory of old industrial sites; and S3IC, the database of classified facilities.

With abundant information about industrial sites, these databases are proven to be necessary for the assessment of the situation of a polluted site and the calculation of the cost for rehabilitating a wasteland. Nevertheless, the information contained in them can become inconsistent across databases due to their specific objectives and different update rates. The BASIAS database has been created to record the activities of old industrial sites. Comparing to other databases, it specializes at classifying the productive activities of a site, but in the meantime some information, for example the address of a site, may not be up to date in this database. The S3IC database has been constructed through inspection of industrial facilities, which means it contains the information about the operations of facilities on a site, the authorization status for the operations and the danger level of those facilities. It classifies the industrial activities conducted through those facilities from the point of view of an MTES inspector, which makes S3IC different from other databases about polluted sites. Lastly, BASOL focuses on the pollution of industrial sites. In this database, each site is described in details through the potential pollution processes and/or the remediation processes, as well as a list of pollutants detected in the site, all of these are missing from the other databases.

The multiplication of databases and their content variations make it difficult to have a synthetic view of the situation of the sites. In addition, historical information such as industrial events also plays an important role in the assessment of sites, but this information is either missing or disorganized in these databases.

Therefore, we have planned to create a memory of sites that reorganizes the information from these databases in a more invariable and efficient way. A memory of sites is a database constructed on events that happened in those sites. Eventually, users will be able to query this database for polluted site information, like location, pollutants and industrial activi-

[a] https://orcid.org/0000-0003-3232-8177
[b] https://orcid.org/0000-0001-7062-0262
[c] https://orcid.org/0000-0002-0362-6805

ties etc. Since the existing databases do not share the same objectives regarding the pollution treatment nor the same definition of an industrial event, they do not record the same events. Besides, those events are usually embedded in narrative texts as a part of databases, and there are a lot more events described in the texts, like regulatory reports, rather than in databases. So, in this paper, we introduce an information extraction model which enables event-related extraction from a plain text. In the future, the chronological assembly of these events will make it possible to build the memory of polluted sites.

The information extraction model suits the BASOL narrative texts from which events must be extracted. So, after a brief introduction about related work in section 2, section 3 describes the narrative text corpus, the notion of event and the features which describe industrial events and are looked for in the corpus. The automatic annotation process is based on deep learning; it combines a neural network and word embeddings; they are explained in section 4. The automatic annotation of the event features is assessed in section 5. The results are discussed, based on precision, recall and F-score measures in section 6. The paper concludes with perspectives in section 7.

2 RELATED WORK

In natural language processing (NLP), an information extraction task can be regarded as a sequence labeling task or a classification task. Information extraction tasks focused on event features are relatively new to the NLP community. Over the last decade, several approaches have been proposed by different researchers. In (Arnulphy, 2012), a machine learning model has been used to classify the words by their predefined syntactic, morphologic, semantic and lexical features in order to recognize the events. This classifier is based on a decision tree algorithm, and eventually gets a 0.74 F-score on linguistic feature classification. In (Battistelli et al., 2013), a data mining approach has been proposed, which involves extracting semantic patterns of sentences that describe an event. So, sentences with similar patterns can be extracted as events. Although these approaches are different in usage of models and algorithms, they all require the assistance of an abundant linguistic resource. For example, in (Arnulphy, 2012), French lexicons including action verbs and event nouns are used to define the lexical features of words. In recent years, the development of artificial neural network and language models has made the deep learning approaches much more viable for NLP tasks, including sequence labeling tasks. In (Panchendrarajan and Amaresan, 2018), a model trained on Bi-LSTM neural network has gained a 0.90 F-score on named entity annotation. In the work of (Shin et al., 2020), a spatial information extraction model based on BERT (Bidirectional Encoder Representations from Transformers) is presented. By implementing the language model BERT, the authors have successfully extracted different types of spatial entities with a F-score of 0.90 in total. From these works, it can be seen that the usage of artificial neural networks and language models has improved the result in sequential labeling tasks, especially semantic annotation, without implementing extra linguistic resources.

Our project to build a memory of polluted sites focuses on extracting information about industrial events. As named entity extraction, event extraction is also a semantic annotation task. Different from previous work, we seek to extract events with a certain theme: pollution. This means that we need an approach with strong ability to process semantic features in text. Our proposed approach is inspired by recent work and is based on a deep learning method and a language model.

3 THE BASOL CORPUS AND THE INDUSTRIAL EVENTS

BASOL describes polluted or potentially polluted sites, and soils requiring preventive or remedial action by public authorities through a structured database of the industrial events, which is complemented by narrative texts. The description of industrial event includes specific features, which are relevant in the context of pollution. The corpus extracted from the BASOL database is first presented. The concept of industrial event with its characteristics is based on this corpus; the design of the labels of the characteristics and their use are then introduced.

3.1 Description of the Corpus

BASOL contains structured information about more than 7 000 polluted sites since the 1990s, including their geographic location, owners' identity and detected pollutants. In addition, narrative texts are added to the database records and provide detailed information concerning the facilities and the industrial sites. The texts collected as a corpus provide the source in which industrial events are looked for. The corpus contains 155 587 sentences, with a vocabulary of 48 032 words. The descriptive texts are meant to clarify the industrial incidents that had an influence on

the site, so they include mentions of industrial events. The vocabulary is focused on the topic of the industrial pollution. Since this is an official database, the usage of standard French is also a significant quality. As an example, the following sentence is taken from the corpus: *La société BRODARD GRAPHIQUE était installée depuis 1959 sur la zone industrielle de Coulommiers* (BRODARD GRAPHIQUE was established since 1959 on the Coulommiers industrial area).

3.2 The Concept of Event

The corpus details industrial events. But what exactly is an event? By the definition of dictionary, an event is "*a thing that happens, especially something important*"[1]. Various definitions of an event have been made in previous works. In her doctoral thesis, (Arnulphy, 2012) defines an event as something happens that changes the state. In (Lecolle, 2009), an event is regarded as a singular, unexpected and unrepeatable case. In (Battistelli et al., 2013), although there is no clear definition of *event*, the importance of date in event extraction is emphasized which implies that *event* is a notion with significant temporal properties. From these definitions, it is shown that *event* is a relatively subjective notion which can be adapted to the need of research. But there is a consistency in these definitions. It is clear that the notions of "important" and "happen" are crucial. These notions represent two major aspects of an event: occurrence and importance. From a semantic perspective, occurrence can be interpreted as having a distinctive and closed time range. And importance implies an impact on the reality. Therefore an event can be defined as something that impacts the reality, with a distinctive and terminated time marker.

In this project, we specifically study industrial events. Based on the definition of event, an industrial event can be defined as something impacts the industrial situation, with a distinctive and terminated time marker. According to this definition, several elements must be defined to specify an industrial event. First, to describe the occurrence of an event, a time marker, an action and an actor are required. Since eventually the events will be linked to industrial sites in the database, a place marker is also crucial. With these elements extracted, we can describe the occurrence of an event as "Who did What When and Where". Second, the importance of the event needs to be described. Although the impact on industry can not be extracted directly from a text, information may be found on the influence of an industrial event on the environment. To gather this information, elements such as pollutants, chemical components and products should also be extracted.

[1]https://www.oxfordlearnersdictionaries.com/definition/american_english/event

3.3 Label Design and Application

Therefore, we propose the following set of labels to designate the features of an industrial event:

- **O**: an object, a nominal phrase that serves as an argument of an action. It can be either the actor, the receiver or the complement of an action;
- **N**: an action trigger of an event, usually a momentary verb or its nominal derivation;
- **A**: an industrial activity; An activity is a repeating action that a company conducts daily;
- **T**: an indicator of time, typically a date;
- **L**: an indicator of location, only geographic and administrative locations;
- **R**: a relation, usually a prepositional phrase indicating the logical relation between other labels;
- **I**: an institution's name;
- **S**: a chemical element;
- **U**: a pollutant other than chemical elements;
- **D**: a pollutant in form of a container for other pollutants, for example a wasteyard.

These labels, while covering the need for annotating basic information, may cause a problem of overlap. For example, in this segment that describes an industrial activity, *aspersion de Xylophène sur les poutres de bois* (in English: Xylophene sprinkling on the wooden beams), label **U** should be assigned to the chemical product *Xylophène* (Xylophene), while another label **A**, industrial activity, is assigned to the whole segment. In order to reduce the risk of overlapping, the labels have been separated into 2 groups. The first one contains the labels **O**, **N**, **T**, **A**, **L** and **R**, which are useful to describe an event or an activity. The **I**, **D**, **S** and **U** labels are in the second group; they provide complementary information about pollution and institution. From a linguistic perspective, the labels of the first group have a strong link to syntactic features of words. The assignment of the first group labels requires information about the part-of-speech and the dependency relations between words, such as whether the word is a noun or a verb, whether it is the subject or the predicate in the sentence. The second group is more related to semantic features, and it is by knowing the meaning of the words that these labels can be assigned. For example, *Hydrocarbure* (Hydrocarbon) is identified as a chemical substance (label **S**)

not because it is the subject of a sentence, but because it means *an organic compound consisting entirely of hydrogen and carbon*[2]. In addition, a priority rule has been defined in order to assign only one label to each word. For example, a place name, annotated as a location, L label (first group), may also be annotated **O** (second group) as the object of an event trigger verb. The rule which has been implemented priorizes the indicator of location, which much more specifies the event than the fact it is an object too.

On the other hand, the designation of the event features are often made up of several words, for example: *La société BRODARD GRAPHIQUE*, *sur la zone industrielle de Coulommiers*. Therefore, the "B-I-E-O" (begin, inside, end, outside) annotation format has been implemented in the annotation work. Since this format uses different labels for the beginning and the end of an extracted expression, it enables to detect multiword units. In this way, both category labels and boundary labels can be assigned at the same time to each word in a group. So, it is easy to distinguish between groups of words, even if they are of the same category. Consequently, the labels assigned to each word is in fact a combination of a boundary label and a category label. Here is an example:

Les	*installations*	*de*	*l'usine*
BO	**IO**	**IO**	**EO**

ont	*été*	*démolies*	*entre*	*1970*	*et*	*1980* .
BN	**IN**	**EN**	**BT**	**IT**	**IT**	**ET**

The two-character labels enable to delimit three phrases: *Les installations de l'usine* (label **O**), *ont été démolies* (label **N**), and *entre 1970 et 1980* (label **T**).

As can be seen in this example, the assignment of labels is realised within a sentence. Normally, the boundary of a sentence does not necessarily match the boundary of an event; some features of an event may appear in a different sentence from the one that contains the trigger of the event. However, the BASOL corpus is a combination of brief texts that summarize the activities and events that occur at a site. So, it is more unlikely to find an event announced in two sentences in this corpus. Consequently, the narrative texts have been segmented and annotated into sentences. This has several advantages. The sentence is a perfect unit for the input of a deep learning algorithm (see the next section), since a paragraph as a unit may be too voluminous for the algorithm to run efficiently, and a word as a unit risks loosing context features of the word. The segmentation into sentences enables a better control of the manual annotation workload.

[2]https://en.wikipedia.org/wiki/Hydrocarbon

4 AUTOMATIC ANNOTATION OF EVENT FEATURES

The targeted memory of polluted sites is based on a chronological assembly of pollution events. Each of them is described through its features; the goal of the information extraction model is to automatically identify and annotate the features. The model which is proposed combines a neural network to identify the phases, and word embeddings to distinguish between the use contexts of each word occurrence. The two components are independent and the choice of each one is guided by criteria that are explained. The training of the model combines both components and is based on the training corpus that has been manually annotated.

4.1 Choice of the Information Extraction Model

Several models are suitable to automatic information extraction. The most adopted ones are the models based on linguistic rules, and those trained with supervised deep learning method.

The rule-based models can perform a very precise information extraction. However, they rely on implemented vocabularies and their performance may deteriorate when processing a corpus with new terminologies, which is known as an Out-of-Vocabulary problem (OOV). This could be a major drawback in our case because the corpus could be extended to other documents that deal with the same theme but with another vocabulary (more technical or more regulatory) or with the mention of new institution names and other chemical product names. Finally, we choose to make a neural model based on deep learning method, in order to solve OOV and to obtain a more flexible tool. The supervised deep learning method on which the model is made is called Bi-LSTM (Bidirectional Long Short-Term Memory) (Basaldella et al., 2018). LSTM is a recurrent neural network (RNN). Comparing to other neural network structures, RNN is more suitable for sequential learning task, especially in the case where the output of an input can be influenced by the previous inputs. This property of RNN suits the feature annotation since a word's label assignment is strongly influenced by the words in its context. Derived from the traditional RNN, the Bi-LSTM neural network is more flexible than RNN in sequence tagging tasks because of its ability of reserving the influ-

ence of a word's remote context during training. And since this is a bi-directional model, it can learn from both previous context and following context, and thus it is more suitable for detecting the beginning and the end boundaries of an expression.

4.2 Choice of Word Embeddings

For text data being able to be processed by the neural network, one step is indispensable: word embedding. Indeed, every input text word is substituted with its vector that the algorithm can process. So, the vector returns the context of the word in the text. Several word integration models exist, which influence the performance of the information extraction models. At the beginning of the implementation, in order to quickly test the performance of Bi-LSTM neural network, we have tried training with one of the simplest word embedding method: Word2vec (Mikolov et al., 2013). This method, while able to efficiently provides word vectors generated from the context of each word, has some fundamental flaws that influenced the performance of the models. First of all, the vector generated by Word2vec is static, this means each word form has one and only one vector for the whole text unit, regardless of its different contexts. Consequently, the word vectors generated by Word2vec model cannot represent polysemy, the case where a word can have different meanings in different context. Furthermore, unlike multi-layer deep learning word embedding models, Word2vec cannot generate vectors that embed complex linguistic information of different levels, such as a word's syntactic and semantic features. Therefore, other word embedding models have been taken into consideration, specially some state of art language models. Finally, we have decided to use the French language model CamemBERT (Martin et al., 2020), a Transformer-based model trained on a large French corpus. This model is known for its state-of-art performance for natural language processing tasks in French, including part-of-speech tagging, dependency parsing and named entity recognition. What makes this model special is that it assigns different vectors to different occurrences of the same word, according to the contexts. And for words it cannot recognize, it breaks down the words into morphemes to assign them the corresponding vectors. Thus, this model is not affected by polysemy or OOV problems. Since this model can efficiently integrate the semantic features in the context, it would be helpful for recognizing the labels closely related to word sense, the pollutants for example.

4.3 Training and Validation Corpora

As explained above, the proposed model is based on a neural bi-LSTM model. It must be trained with an annotated corpus in order to learn the labels which annotate the event features. The annotated corpus must be reliable (annotations must be manually checked), consistent, suitable for the task and of sufficient size. In addition, a part of the manually annotated corpus must be reserved for the assessment task. In order to reduce the manual annotation work, a "bootstrapping" annotation-training process has been implemented. First, the event-related information is manually annotated in a small sample of corpus. Then, the model is trained on this annotated sample to become a rough trained annotation model. Through this model, another corpus sample can be automatically annotated and then manually corrected, resulting in a new training cycle for the model, which improves it. By repeating this process we can perform a "bootstrapping" annotation-training process. It enables to accumulate annotated and checked samples which are gathered to form the final training corpus. Thus, the model can be trained, as much as necessary, on an abundant and reliable corpus and become an efficient tool.

As seen before, the narrative texts have been segmented into sentences and annotated. Thus, each input data unit of the model is a sentence which is in the form of a tensor that contains the vector of every sentence word.

The passage from a sentence to its words is based on a tokenization process. To ensure the coherent combination of the different components of the final model, the tokenization method of the word embedding provider, i.e. CamemBERT, has been adopted. However, the way that CamemBERT splits certain words into lexemes can cause inconvenience for manual annotation or correction. Therefore, a script that can transform the CamemBERT tokens to TreeTagger (Schmid, 1994) tokens[3] has been prepared, along with their labels. The TreeTagger tokenization is the one chosen for the manual annotation, but this script can also transform CamemBERT tokens to any other types of tokens. The script can also work in the opposite direction, and transform other tokenized sentences to CamemBERT tokens.

This is a bootstrapping experiment that augments the annotated text through the model training sessions. For the first session, only 120 annotated sentences were prepared for training the model, and 100 sentences to test and evaluate it. After applying the model, we manually corrected the annotation result,

[3] https://github.com/DongChuanming/KDIR_2021_shared/blob/main/KDIR_tokenization_transformer.py

and thus obtained 100 more correctly annotated sentences.

The second session has consisted of several steps: first, a transitory model has been trained on the 220 annotated sentences, then by using this model, 301 new sentences have been automatically annotated. This enables to efficiently obtain 301 more parsed sentences by correcting the annotation result. Then these sentences have been split into 3 groups: 130 sentences join the training data, giving 350 sentences for model training; 120 sentences for developing, more precisely for choosing the number of epochs; and the evaluation set composed of those 120 sentences complemented with the last 51 annotated sentences.

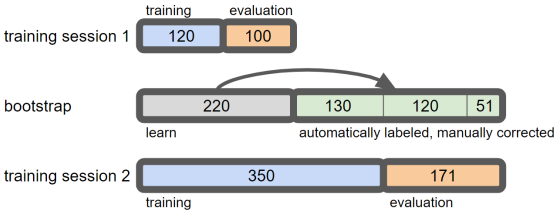

Figure 1: Illustration of the bootstrap method used to augment the training and evaluation corpora.

5 EVALUATION OF ANNOTATION MODELS

Since the labels have been separated into two groups, two models (named Model 1 and Model 2) have been implemented to automatically annotate the event features. Both are based on the Bi-LSTM neural algorithm and use the same word embeddings provided by CamemBERT. They have been trained and assessed with the same training and evaluation corpora. They share the same training processes (numbers of epochs and learning rate), named session below.

During model training, the evaluation has already begun. In order to find the parameters that optimize the training, we have tested the models with 120 developing sentences with different network configurations. To illustrate, here is a graph that shows how the F-score of each label of Model 1 evolves according to different numbers of epochs, with learning rate at 0.01 :

According to figure 1, at epoch 400, most labels have the highest F-score, thus 400 is the best epoch number for Model 1 training if other parameters don't change. Aside from epoch number, we have also tested other parameters like learning rate and butch size, for both Model 1 and Model 2, to find their best value. The evaluation results presented below are for

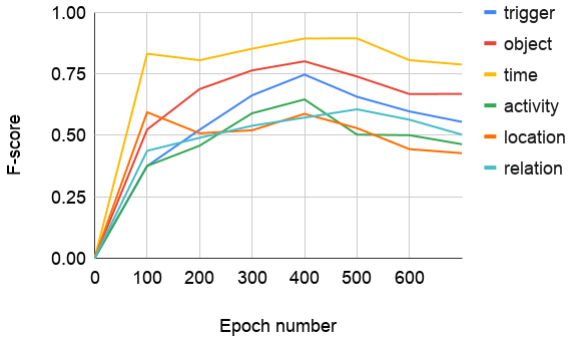

Figure 2: Evolution of the F-score computed on the developing set of the first session, by epoch number for each label - Model 1.

models trained with the best parameters at the moment. Since all parameters have not yet been tested, it is possible that the models will be further improved. The evaluation results of the two models trained during both sessions are shown in the following tables. The evaluation is realised on each label separately. Since event-related information has been extracted through the category labels, at this stage, the boundaries labels have not been evaluated. Table 1 and 2 are the evaluation of Model 1 and Model 2 trained during the first session.

Table 1: Number of true positives (TP) and evaluation of the precision (p), recall (r) and F-score of Model 1 on the test set of the first training session (100 sentences, 400 epochs).

Label	TP	p	r	F-score
trigger (**N**)	143	0.66	0.57	0.61
activity (**A**)	64	0.40	0.58	0.47
object (**O**)	209	0.94	0.85	0.89
time (**T**)	184	0.93	0.88	0.90
location (**L**)	61	0.63	0.59	0.61
relation (**R**)	23	0.45	0.45	0.45
Total	684	0.72	0.70	0.71

Table 2: Result of Model 2 on the test set of the first training session (100 sentences, 400 epochs).

Label	TP	p	r	F-score
institution (**I**)	28	0.93	0.46	0.62
chemicals (**S**)	29	0.88	0.58	0.70
pollutant (**P**)	2	0.10	0.40	0.16
container (**D**)	0	-	-	-
Total	59	0.71	0.50	0.59

Table 3 and 4 show the evaluation of Model 1 and Model 2 trained during the second session.

Table 3: Result of Model 1 on the evaluation set of the second training session (171 sentences, 400 epochs).

Label	TP	p	r	F-score
trigger (**N**)	308	0.77	0.69	0.73
activity (**A**)	120	0.62	0.64	0.63
object (**O**)	763	0.89	0.82	0.85
time (**T**)	194	0.89	0.93	0.91
location (**L**)	86	0.55	0.63	0.59
relation (**R**)	157	0.61	0.54	0.57
Total	1628	0.78	0.74	0.76

Table 4: Result of Model 2 on the evaluation set of the second training session (171 sentences, 400 epochs).

Label	TP	p	r	F-score
institution (**I**)	146	0.95	0.77	0.85
chemicals (**S**)	95	0.90	0.82	0.86
pollutant (**P**)	54	0.75	0.47	0.57
container (**D**)	2	0.25	0.15	0.19
Total	297	0.87	0.68	0.77

6 RESULT ANALYSIS

Although we only used a small manually annotated corpus, we already obtained promising results on the models. For a simple comparison, we have also tested two other NLP tools on date annotation, a popular Python library called `dateparser`[4], and `NOOJ`[5], an annotation software for linguists. Both of them are based on rules. Considering the reliance of event on its time marker, this comparison should be able to reflect the performance on event extraction too. As a result, `dateparser` can only detect the date expressions in our text with a 0.77 precision and a 0.48 recall; `NOOJ` obtained 0.98 precision, but only a 0.44 recall. This proves that our models have a state-of-art performance for detecting certain entities. By observing the score of the different labels, and by comparing the manual and automatic annotations, we have discovered some interesting points to address. The score of the different labels, and the comparison between the manual and automatic annotations give clues to improve the results of the automatic annotation of the event features. The commentaries of the results and the improvement clues are organized regarding three themes : the confusion between labels, the improvement due to the increase of the corpus, and the relevance of the CamemBERT word embeddings.

[4]https://dateparser.readthedocs.io/en/latest/

[5]http://explorationdecorpus.corpusecrits.huma-num.fr/nooj/

6.1 Comparison between Labels

The models do not work well on some labels. Comparing to time (**T**) and object (**O**) labels, event trigger (**N**), industrial activity (**A**) and location (**L**) labels do not have an impressive F-score. After observing the automatic annotation results on these labels, we see that certain sentences that should have been annotated as event trigger, are annotated as industrial activity. Based on our definition of event trigger, the action that triggers an event should be a momentary verb or its nominal derivation. In contrast, an industrial activity is an action conducted by enterprises frequently during a period of time, and should be designated by a durative verb or its nominal derivation, or a repeating action. However, it is difficult to distinguish an event trigger expression from an industrial activity expression, based on their syntactic features, especially when they are all nominal derivation of verbs. Unlike a verb, a noun does not have "momentary" nor "durative" as properties. Therefore, once nominalized, these event trigger expressions are confused with an activity, usually in the form of a nominal phrase.

A similar problem can be found with the label *location*. Since the expression of a location often has a prepositional structure, the nominal part of a location expression can easily be recognized as an *object* if its position is close to an event trigger or an activity. Besides, based on its definition, the recognition of a location expression is trickier. The location expressions we want to extract include only geographic and administrative locations. For example, even if the prepositional phrase *dans les nappes des calcaires grossiers* (in the coarse limestone sheets) indicates a position and hence is annotated by our model as a *location*, it does not belong to either precedent types , and therefore should not be recognized as a location.

6.2 Improvement Due to the Corpus Increase

An improvement can be observed between the two sessions. Comparing to the first session, the models trained in second session have a better performance on annotating most labels due to the increase of the training text. Also, it is noticeable that Model 2 has benefited more from this training corpus increase. Indeed, the labels of the second group are less frequent than those of the first group. Consequently, there are not enough second group annotation examples in the first session; the category *container* (**D**) is even absent from the test corpus of the first session. With more training text attached to the second session, the models are able to learn the second group annotations

on more label instances and thus improve Model 2.

6.3 Relevance of the Word Embeddings

The use of the CamemBERT word embeddingds also improved the results. The *pollutant* category (**P**) is the one that benefits the most from the use of the vectors. To compare, by using our preliminary model implementing the Word2vec method, the pollutant annotation precision is only 0.05 but by using the current model the score has increased to 0.56 without lowering the recall. A pollutant expression is usually a nominal phrase. It is very difficult to differ it from any other nominal component, on syntactic level. And unlike institution names or chemicals, the expression of pollutants does not involve changes of word case or the usage of nomenclatures. So the most promising ways to recognize them are by analysing the polarity (positive or negative) in the context, and by building the word meaning itself, all of which require usage of complicated semantic features. Unlike syntactic features, semantic features are hard to extract and to be comprehended by the algorithm. The CamemBERT model, which has embedded semantic features in form of word vectors, enables the neural network to learn annotation patterns on a semantic level. So, our model can recognize some typical pollutant expressions, like *tensio actif* (surfactant) and other chemical products, which is exactly the information which must be extracted in order to build the memory of polluted sites.

7 CONCLUSION

In this paper, we have described an approach for event-related information extraction from a corpus focused on industrial pollution. With a supervised deep learning method, we trained two models that can simulate our manual annotation on industrial event features. Right now, the models trained with Bi-LSTM neural networks have given promising results, but we still need them to be better at detecting event triggers and industrial activities in order to use them on other text resources. Given the fact that the models are trained with only a small portion of the corpus, and the neural network configurations are not fully explored, it could be possible to improve the model. Aside from increasing training text data and adjusting neural network setting, it is also interesting to see if the model could have a better performance if we use paragraphs instead of sentences as the input of the neural networks, since the narration of an event is not limited in a sentence.

This work is devoted to the construction of the polluted sites memory, based on an only consistent and complete database. Eventually, the event-related information extracted by the models will be inserted in the database. For future work, we will apply a syntactic parser to link the extracted event features by dependency relations, and train a classifier to categorize the events, so that they can be integrated into the database with an appropriate structure. The models will also be tested and used on other corpora in the domain of industrial pollution, to connect other sources of data and enrich the polluted site memory.

REFERENCES

Arnulphy, B. (2012). *Désignations nominales des événements: étude et extraction automatique dans les textes*. PhD thesis, Université Paris 11.

Basaldella, M., Antolli, E., Serra, G., and Tasso, C. (2018). *Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction*, pages 180–187. Springer.

Battistelli, D., Charnois, T., Minel, J.-L., and Teissèdre, C. (2013). Detecting salient events in large corpora by a combination of NLP and data mining techniques. In *Conference on Intelligent Text Processing and Computational Linguistics*, volume 17(2), pages 229–237, Samos, Greece.

Lecolle, M. (2009). Éléments pour la caractérisation des toponymes en emploi événementiel. In Evrard, I., Pierrard, M., Rosier, L., and Raemdonck, D. V., editors, *Les sens en marge Représentations linguistiques et observables discursifs*, pages 29–43. L'Harmattan.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., and Sagot, B. (2020). CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Panchendrarajan, R. and Amaresan, A. (2018). Bidirectional LSTM-CRF for named entity recognition. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees.

Shin, H. J., Park, J. Y., Yuk, D. B., and Lee, J. S. (2020). BERT-based spatial information extraction. In *Proceedings of the Third International Workshop on Spatial Language Understanding*, pages 10–17, Online. Association for Computational Linguistics.