Iterative Learning for Semi-automatic Annotation Using User Feedback

Meryem Guemimi and Daniel Camara

Center for Data Science, Judiciary Pôle of the French Gendarmerie, Pontoise, France
meryem.guemimi@gendarmerie.interieur.gouv.fr
daniel.camara@gendarmerie.interieur.gouv.fr

Abstract. With the advent of state-of-the-art models based on Neural Networks, the need for vast corpora of accurately labeled data has become fundamental. However, building such datasets is a very resource-consuming task that additionally requires domain expertise. The present work seeks to alleviate this limitation by proposing an interactive semi-automatic annotation tool using an incremental learning approach to reduce human effort. The automatic models used to assist the annotation are incrementally improved based on user corrections to better annotate the next data. Labeling efforts can be largely reduced as reviewing annotations is faster than reading unannotated text and looking for a sequence of tokens to annotate. To demonstrate the effectiveness of the proposed method we build a dataset with named entities and relations between them related to the crime field with the help of the tool. Analysis results show that annotation effort is considerably reduced while still maintaining the annotation quality compared to fully manual labeling. We use the constructed corpus to train multilingual Named Entity Relation extraction (NER) and Semantic Relations Extraction (SRE) models and compare their performance to that of the final evolutive models generated during the annotation.

Keywords: Semi-Automatic Annotation, Natural Language Processing, Named Entity Recognition, Semantic Relation Extraction, Evolutive Model, Incremental learning, Criminal Entities.

1 Introduction

The explosion of digital data in the last decades resulted in an exponential increase in structured and unstructured information with a massive growth for the latter. Unstructured data either does not have a predefined data model or is not organized consistently, contrary to structured data that presents a format, which improves its usability.

According to Computer World [1], unstructured information may account for more than 70% to 80% of all data in corporations. For many organizations, appropriate strategies must be developed to manage such volumes of data. This is the case for general companies, but also intelligence agencies. The Central Service for Criminal Intelligence (CSCI) of the French Gendarmerie receives and processes thousands of documents per year. Only in terms of formal complaints the CSCI receives ~1.8 Million each year. The

information comes from different sources and in a significant part in unstructured form. As an example of documents, we can cite criminal reports, signaling from citizens and companies. These documents, sent by heterogeneous and voluntary sources, make it impossible to impose or even control the formats of the reports. However, having structured information is crucial for investigators and intelligence analysts who spend a considerable amount of time analyzing this data. Hence it is crucial to develop techniques that automatically organize text in a structured way such that the information obtained can be directly analyzed, classified, and used by other, higher-level information management tools.

State-of-the-art text mining tools are based on Deep Learning techniques that require sufficiently large corpora of labeled data. The unavailability of such resources and the prohibitive cost of creating them are tackled in this paper. Today we may find different frameworks that proposed pre-trained models that cover generic named entities and relations, the creation cost of these models is shared among a large set of users. However, such pre-trained models lack domain-specific knowledge, and for many fields, the available answers are not precise enough. Law enforcement is not an exception. The vocabulary of the analyzed documents and the named entities and relations of interest vary significantly from those proposed by the regular pre-trained models. In this situation, transfer learning or even the full retraining of available models may be required. The retraining process implies the annotation of a fair number of documents where the domain-specific vocabulary and named entities are treated.

Our goal is to create a tool that simplifies and speeds up the annotation process for different categories of text analysis tasks. In this paper, we investigate the effectiveness of the proposed method on the Named Entity Relation extraction (NER) [2] and Semantic Relations Extraction (SRE)[3] [3]. However, the idea can be applied to a larger set of tasks. At first, we reuse standard pre-trained models capable of extracting general named entities and relations between them. As the user provides domain-specific text, the tool pre-annotates it providing broad categories, such as people, organizations, and locations. The user is asked to review the result and, if necessary, add missing domain-specific entities/relations. This update triggers a background training of the automatic models using the knowledge injected by the user. After several iterations, the models learn and start proposing the user's domain-specific categories. The model's accuracy becomes high enough that we switch from an annotation mode to a reviewing mode at a later stage. This simplifies and reduces the amount of manual labor and level of expertise required to annotate domain-specific texts. The tool was presented and made available for EUROPOL to assist and ease their work in annotating entities and relations in text.

The remainder of this paper is organized as follows: Section 2 gives a brief overview of some related work. In Section 3, we outline our pipeline proposal, frameworks used, and experimental setup. Section 4 reports and analyses our results and Section 5 concludes the paper.

2 Related Works

Machine learning methods provide fundamental advantages and state-of-the-art results but still require a large amount of labeled data to learn. Nonetheless, annotated corpora are often expensive to produce, leading to a deficiency of labeled datasets for specific domains and low resource languages. Since manual labeling of data is a costly, labor-

intensive, and time-consuming task, researchers have been exploring techniques to derive better annotation methods that minimize human effort. Different techniques have been proposed to partially or fully automate the annotation process to cope with the high demand for annotated training corpora without relying on manual annotation. In this section, we present some of these studies. No attempt is made to be exhaustive, as the goal is to compare and contrast these with our efforts. The approaches that these projects take can be categorized into semi-automatic, and automatic generation approaches.

2.1 Semi-automatic approaches

Semi-automatic text annotation has been the subject of several previous studies. It combines automatic system predictions with human corrections by asking a human annotator to revise an automatically pre-tagged document instead of annotating it from scratch.

In their study, Komiya et al. [4] show that this approach can significantly improve both annotation quality and quantity. They compare manual annotation to a semi-automatic scheme where non-expert human annotators revise the results of a Japanese NER system. This method reveals that the annotation is faster, results, on average, in a better degree of inter-annotator agreement and higher accuracy. Following this line of work, Akpinar et al. [5] conduct a series of experiments to measure the utility of their tool and conclude that this approach reduces by 78.43% the labeling time, accelerates the annotators learning curve, and minimizes errors compared to manual tagging. Ganchev et al. [6] take a similar approach but with a different implementation that only allows binary decisions (accept or reject) from the human annotator. They conclude that this system reduces the labeling effort by 58%.

Halike et al. [7] point out the utility of this approach for low resource languages. Their work expands an existing Uyghur corpus with Named Entities and Relations between them using a semi-automatic system. Their method enables rapidly building a corpus and training a state-of-the-art model tackling the deficiency of annotated data.

Cano et al. [8] present BioNate-2.0, an open-source modular tool that comes with a collaborative semi-automatic annotation platform allowing the combination of human and machine annotations. Their pipeline includes corpora creation, automatic annotation, manual curation, and publication of curated facts. Different projects adopted the tool and the authors plan to further improve it by implementing an active learning method to prioritize the annotation process and further reduce curations of the human annotator. Neveol et al [9] study the efficiency of a semi-automatic tool to build a new labels corpus of biomedical queries. They conclude that this approach is beneficial to assist large-scale annotation projects as it helps speed up the annotation time and improve annotation consistency while maintaining a high quality of the final annotations.

This approach is generally found helpful by most annotators; however, it still requires human intervention and is not efficient when applied to specific domains far from which the automatic model was trained, as Komiya et al. [4] demonstrated. Thus, requiring an initial manual annotation to help increase efficiency.

2.2 Semi-Automatic with Iterative Learning approaches

Other researchers take this idea one step further by proposing a semi-automatic approach with an interactive system that incrementally learns based on user feedback. The component used to tag the data automatically is updated at regular rounds based on user corrections to increase its efficiency and reduce the number of annotator updates.

Wenyin et al. [10] use this strategy for Image Annotation via keyword association for image retrieval. Their strategy is to create and refine annotations by encouraging the user to provide feedback while examining retrieval results. When the user provides some feedback about the retrieved images, indicating which images are relevant or irrelevant to the query keywords, the system automatically updates the association between the other images based on their visual similarity. The authors conclude that through this propagation process, image annotation coverage and quality are improved progressively as the system gets more feedback from the user.

Bianco et al. [11] develop an interactive video annotation tool integrating an incremental learning framework. This approach is implemented on the object detection module, enabling expanding the domain knowledge and iteratively increasing its efficacy. Results demonstrate that the system reduces the average ratio of human intervention of at least one order of magnitude with respect to the complete manual annotation mode while preserving the annotation quality. The authors also highlight the utility of such systems for large-scale annotated dataset generation.

Even though these studies were not applied to textual information, they provided some valuable guidelines for the development of our work, such as the suggestion to keep the ontology simple and the need to support annotators with interactive GUIs. This paper proposes a similar method to annotate general and crime-related entities and relations in free text. We present a semi-automatic text annotation tool that iteratively updates auxiliary NLP models based on user feedback. The platform comes with a Web-based User Interface to enable a user-friendly annotation correction and ensures continuous communication with a server responsible for training and improving the NLP models. Unlike the previously described studies, we additionally evaluate the impact of the model update frequency on the annotation and compare the intermediate models to a traditionally trained model, i.e., once with all the labeled dataset.

2.3 Fully automatic approaches

While these techniques require a significant amount of human labor, less than manual annotation but still considerable, other studies focused on fully automatic annotation.

Laclavik et al. [12] present Ontea, a platform for automated annotation based on customizable regular expression patterns for large-scale document annotation. However, as the authors mention, the success rate of the technique is highly dependent on the definition of the patterns, but this solution could very powerful for enterprise environments where business-specific patterns need to be defined and standardized to identify products. Similar to this work, Teixeira et al. [13] and Hoxha et al. [14] propose a method to construct labeled datasets without human supervision for NER using gazetteers built from Wikipedia and news articles. The evaluation results show that the corpora created can be used as a training set when no other is available but still are considered of silver quality and may lead to low performance trained models.

Canito et al. [15] make use of data mining algorithms to annotate constantly flowing information automatically. They test their approach on classification, clustering, and NER. They conclude that this approach is suitable for scenarios where large amounts of constantly flowing information are involved, but the results are poor compared to manual and semi-automatic techniques.

Menezes et al. [16] automatically generate a labeled dataset for NER in Portuguese by exploiting structured data from DBpedia and Wikipedia. The dataset is constructed by tagging tokens from Wikipedia sentences that exactly match a known entity in DBpedia. Additionally, they use an auxiliary NER predictor to capture missing entities. They conclude that this dataset yields a performance boost only when used along with a manually labeled dataset.

These fully automatic methods considerably reduce manual labor but have a lower precision or recall compared to other techniques.

3 Our Pipeline Proposal

In this paper, we propose a semi-automatic tool for textual information annotation that combines the efficiency of automatic annotation and the accuracy of manual annotation. The strategy is to iteratively update and refine the inference models to propagate the knowledge gained from user feedback and reduce human intervention until the whole corpus is annotated. The technology is demonstrated on two Natural Language Processing, namely NER & SRE. Any other tasks could be used within the annotator framework, but they are not the focus of this work.

3.1 Proposed Strategy

We present here an NLP annotator platform based on a Web Interface that sends requests to a REST Server that automatically identifies and tags entities and relations between them in the input text. The human annotator is then asked to correct the model prediction instead of annotating the text from scratch. The key idea is to change the task from a manual annotation to a manual reviewing and use the corrections introduced to retrain the model, making the annotation process much faster and more pleasant.

The motivation behind retraining the model is to propagate the knowledge gained from the corrected documents to the following ones. It is important to notice that this will increase the automatic model precision on known tags while it learns new classes introduced by users. Indeed, annotators may identify during the annotation process, a new class that is of interest. This technique will detect the introduction of these new labels and update the model's architecture accordingly. After a few examples, as the model learns, the new class will naturally start appearing on the following pre-annotated documents.

We believe this approach will further reduce the annotation burden by reusing the knowledge gained through the corrections to annotate the next documents. Additionally, it can help identify potential sources of errors and ambiguities in our entity definitions. Finally, this will enable to revise or correct possible flaws in the annotation guidelines early rather than at the end as in traditional linear annotation methods [17].

3.2 User Interface

There are many widely used annotation tools dedicated to NLP tasks in the literature (BRAT [18], GATE [19]). We examined some of these tools, but free and open-source versions of these platforms did not offer all features required to conduct our study. The functionalities we were looking for were the automatic tagging of NEs and SRs, the possibility to add new classes to the annotation scheme and the possibility to retrain automatic models. Additionally, due to the sensitivity of the dataset used during the experimentation, we decided to design our tool for security reasons.

The tool is based on a lightweight Web interface using a REST API to enable communication between the client and the server. When the server receives a request to annotate plain text, it returns a JSON object with the entities and relations automatically detected with the trained models. The interface displays the input text highlighting detected entities on the annotation view alongside a relational graph constructed with the recognized relations. The platform supports different input and output formats such as spaCy Json, CoNLL-IOB, and CoNLL-BILUO.

The interface was designed to be intuitive and user-friendly to make the annotation process faster and more pleasant. With simple mouse clicks, the user can manually create or remove entities and relations from the annotation view. The update is automatically detected and saved in the dataset. As shown in Fig. 1. the interface divides the screen into three main windows. The annotation scheme creation can be seen on the top, the NER results can be viewed on the bottom left and the relational graph on the bottom right of the page. These annotations can be edited in this same view.

3.3 Annotation Tool Process

A typical user scenario is the following. First, the user prepares a dataset in the supported format and uploads it to the tool. The system automatically sends an annotation request to the server and displays the results. The user can review and correct the pre-annotated document and click on the Next button to repeat the operation on the next document. Next, the system sends in the background a training request to the server with the last document manually corrected. Finally, the server finetunes the automatic models based on the user feedback, stores the updated models to the system, and uses them for the following inference round.

The training process is implicit to users during the daily use of the system. While they correct the annotation manually, the server trains the current model on all previous examples, then uses it to tag the next ones, and so on until the whole corpus is annotated.

As shown during the experiments detailed in section 4.6, this strategy is practical and fairly easy to use, although we are still iterating the design of the user interface through user studies.

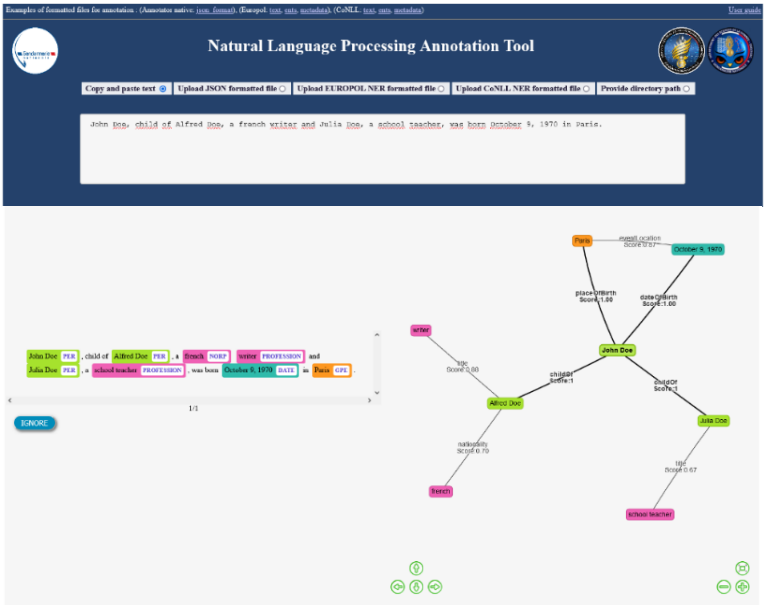


Fig. 1. Natural Language Processing Annotator User Interface. This screenshot shows an instance of a pre-annotated sentence.

Fig. 2. shows the implemented pipeline and how the processes take place at the server side following this order:

- The server tags the provided document using available models
- The user corrects the resulting mentions manually
- The server retrains the model with all previous documents
- The process repeats until all documents are annotated

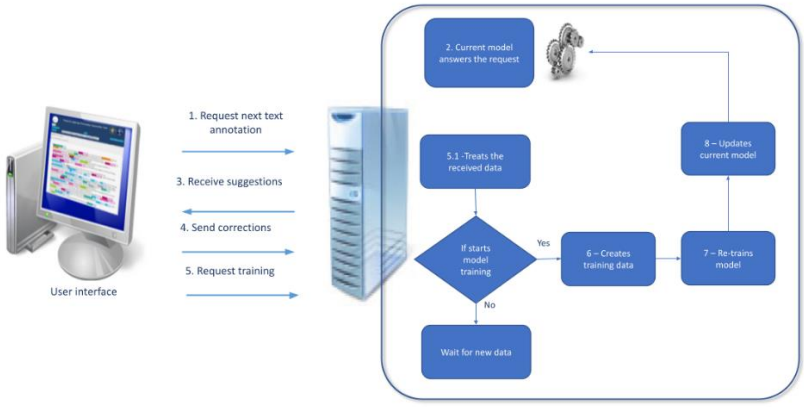


Fig. 2. System architecture showing the annotation and incremental training processes

3.4 Training Process

As explained in Section 4.1, during the annotation process, the system uses generic NLP models pre-trained on large corpora for curation assistance. Different scenarios may arise during this process. The models may be applied to a specific or utterly different domain or genre unknown by the generic models. New use cases or even new classes of tags may be identified. This novelty should be captured by the models to better fit with the data in hand and keep up with changes introduced by the user. This is made possible through transfer learning.

Many practical applications require learning new capabilities while maintaining performance on existing ones. Ideally, the new tasks could be learned while sharing parameters from the old ones without retraining on the whole original dataset through transfer learning. However, this approach may suffer from a performance degradation on old tasks, also known as catastrophic forgetting ([20], [21]). Catastrophic forgetting is a problem faced by many machine learning models and algorithms. When trained on one task, then trained on a second task, many models "forget" how to perform on the first one. This is especially observed for neural network-based systems. Different finetuning adaptation techniques have been studied to reduce this effect and train models on new tasks while preserving their original capabilities [22].

Our case is more straightforward since the old task, i.e., the original set of classes to recognize, can be seen as a subset of the new one that introduces new classes.

The goal is to perform model expansion while still using the original network's information. For this reason, we preserve the original weights of the previous architecture and add new task-specific nodes with randomly initialized parameters finetuned at a later stage instead of training a new model from scratch. To tackle the possible catastrophic forgetting, at each training call, we finetune the model on the latest corrected texts mixed with a random sample of previously reviewed documents. This ensures the training set contains a fair distribution of most of the classes recognized previously by the model along with the newly introduced ones. As the model requires a few examples to learn a completely new class, we use a heuristic for training data construction. The idea is to give a higher weight to the last batches of received documents. It can be seen as a warm-up step to enhances finetuning's on new classes and rapidly converge the new weights.

This shortcut helps saving time, while still getting better performance. We are able to expand the set of possible predictions without the need to retrain the model from scratch and jointly improve the model performance on old classes.

The iterative training approach may be prone to overfitting as the model is re-trained multiple times on repetitive data. However, this statement is not totally unfavorable for our use case as the goal of the incremental process is somehow to mimic the annotator behavior and not train a final model for production. Ideally, if the dataset used is

composed of similar documents, the closer the model gets to the fed documents, the fewer corrections are needed, but this does not apply if the documents are from entirely different domains. To investigate this, we evaluate the model performance throughout the annotation phase against a test set compared to a traditionally trained model. Results of this analysis are reported in section 5.

3.5 Framework selected

As presented above, the annotation strategy continuously retrains prediction models based on the reviewed documents. Many frameworks can be used to assist the annotation process. For this reason, the tool was built in a modular way, so that any other model that supports the training features described above can be integrated. The specific ones we selected were chosen only for our initial investigation of the strategy, as comparing frameworks is not the focus of this study.

Hugging Face.

Named Entity Recognition is a sub-task of NLP that recognizes information units. It consists of tagging entities in text with their corresponding types such as person, organization, location, date, and monetary value. To perform NER, we use a multilingual BERT-based Transformer model, supporting 104 languages. BERT [23] is a pre-trained transformer network [24], which sets for various NLP tasks new state-of-the-art results, including question answering, sentence classification, and sentence-pair regression. It was trained on a large corpus of multilingual data in a self-supervised fashion using a masked language modeling objective. This enables the model to lean an internal representation of the languages in the training set that can then be used to extract features useful for downstream tasks.

We finetune the model on the NER task by adding a token classification head on top of the hidden-states output. Our work is based on the implementation distributed by Hugging Face [25]. However, we modified the original trainer implementation to add new classes to the model architecture on the fly without the need to retrain the model from scratch.

Another worth mentioning feature of this model is that it supports a zero-shot transfer learning approach. BERT's multilingual feature representation allows performing zero-shot transfer from one language to another. This means that if you finetune your model on a specific language, the knowledge gained will be transferred to other languages intrinsically.

BREDS.

Semantic Relationship Extraction is another NLP sub-task responsible for extracting named semantic relationships between entities in a text given some information about the relationships of interest. Extracted relationships usually occur between two or more entities of a certain type (e.g., Person, Organisation, Location) and fall into several semantic categories (e.g. married to, employed by, lives in). For the SRE task, we base our work on BREDS [26], an adaptation of Snowball [27] algorithm that uses word embeddings to compute the similarity between phrases. It is a bootstrapping approach that iteratively expends a set of initial seeds by automatically generating extraction patterns. We choose this method as bootstrapping approaches do not require a large labeled

dataset and can easily scale to other relations by adding new patterns or new seeds. Additionally, this method fits the mental model of investigators that usually have examples expressing a known or unknown relation and aim to find similar seeds and/or discover the nature of the relation.

Bootstrapping approaches usually suffer from a low recall and require manual refinement to achieve higher precision. To overcome this, BREDS makes use of a semantic drift control module to ensure we do not deviate from the initially expressed semantics.

We improved the original BREDS pipeline to expand the extraction to non-verb mediated relations and replaced the monolingual word embedder with a multilingual BERT-based sentence embedding model [28]. It is a modification of the pre-trained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings. It outperforms BERT in the sentence embedding task as BERT computes individual word representations and averages these values for the different tokens of a sentence, resulting in a sentence mapping unsuitable to be used with common similarity measures.

3.6 Experimental Setup

In addition to developing a new corpus of general and crime-related entities and relations between them, this study aims to determine how to best address this task using a semi-automatic annotation tool. To assess the effectiveness of the proposed technique, we perform two different experiments. The first analysis compares semi-automatic to fully manual annotation in terms of speed, accuracy, and user experience. The second analysis studies the influence of the incremental update frequency on the evolutive models and compares their performance to a traditionally trained model.

Experiment 1. Ten annotators with a variety of backgrounds participated in the study. Each user was asked to curate documents manually and using the semi-automatic approach. Both experiments were done using the same annotation tool. To address the proficiency bias, half of the annotators started with the manual mode and the other half began with the semi-automatic mode. For each document they worked with, the total annotation time and editing (adding, removing) actions were saved. Finally, annotators reported their general satisfaction with both experiments by filling a questionnaire.
By assisting curators with automated annotations, we expect their work to be considerably reduced in time and complexity since they have to correct previous annotations rather than create them from scratch. However, due to time constraints, this experiment was performed on only a subset of the dataset. The goal was to get an effort measurement and assess the feasibility of the study. Additionally, we ask an expert annotator to manually label the data used in this experiment to evaluate the generated annotations quality.

Experiment 2. In this experiment, only one annotator was recruited to label the whole dataset. The annotator had been involved in the creation of the local manually annotated corpus and had experience annotating named entities and relations. During the annotation, we trained two different evolutive models, one every time a new document is reviewed and the second one every time ten new documents are corrected, to assess the

impact of the updates on the model's performance. We are also interested in knowing how far these incremental models will be, performance-wise, from the final model trained once on the whole corpus. We suspect that the second approach will be more accurate as it is less prone to overfitting. Results and analysis of the findings of these experiments can be found in section 5.

Before running these experiments, we started by annotating a couple of documents manually with domain experts to get familiar with the task and refine the annotation guidelines. Then, we retrained NLP models on these documents. Finally, once the accuracy of the system became stable, we started the experiments.

Dataset.

The dataset used consists of real, non-confidential, and anonymized documents collected from the internal database of SCRC. The documents were randomly extracted to avoid bias and have a large representation of the SCRC knowledge database. It consists of several texts, including crime reports and complaints. These reports contain a text description of the situation, which often contains several named entities. It is essential to train our model on this specific type of data since text media influences the model behavior. Even if the annotated texts were mainly in French and English the method uses a zero-shot transfer learning technique, which ensures that the results of the annotations can be transposed to different languages.

The corpus provided is composed of 3 different types of documents:

- Modus Operandi: These are short paragraphs with a brief description of a complaint reported by police officers.
- PDF reports: These are large PDF documents of 2 to 4 pages and contain information sheets with several infractions' descriptions.
- GV: They represent short paragraphs describing the cause behind the detention or release of a suspect or witness and the related case.

Annotation Scheme.

To develop the annotation scheme, we first manually inspected with domain experts the type of information contained in the documents and identified entities and relations of interest. Then we started annotating a couple of documents manually to enrich the annotation guideline. During this round, we encountered some special cases and adjusted the annotation guidelines accordingly. For instance, some modus operandi texts do not specify precisely what the infraction is, but it is an information that can be inferred from other elements of the text. For example, the words: 'rummage' and 'break-in' indicate a possible theft. These elements were therefore marked as crime elements (CELM).

Our final annotation scheme for Named Entities is presented in Table 1 and Semantic Relations in Table2.

Table 1. Types and descriptions of entities in the annotation scheme.

Entity Label	Description
PER	person name excluding titles
LEO	law enforcement officer
ORG	companies, organizations, institutions, etc.
NORP	nationalities or religious or political groups.
ADDRESS	full address with street and city or postcode.
POI	point of interest (eiffel tour, airport cdg, etc.)
FAC	buildings, highways, etc.
GPE	geopolitical place names (countries, cities, states).
DATE	absolute or relative dates or periods.
TIME	times smaller than a day.
PERIOD	action duration.
MONEY	monetary values, including unit.
PROFESSION	job titles.
DRUG	medicine or substance referring to a drug.
WEAPON	firearm, cold weapons, etc.
CRIME	infraction.
CELM	crime elements: words or expressions referring to an infraction.
WEB	anything related to web activity (website, social network, cyber activity, etc.).
EVENT	festivals, sports events, etc.

Table 2. Types and sub-types of relations in the annotation scheme.

Relation Label	Sub-types
PER-PER	familial and social relations, aliases, criminal action.
PER-OBJ	NORP (nationality), DATE (date of birth, date of death), GPE (place of birth, place of death), ADDRESS (place of residence), PROFESSION (title), CRIME (victim, assailant)
PROFESSION-OBJ	ORG (Organization affiliation), GPE (place of work)
ORG-GPE	physical location
DATE-TIME	timeline
CRIME-OBJ	DATE (crime date), TIME (crime time), GPE (crime location), MONEY (damage)
EVENT-OBJ	DATE (event date), GPE (event location)

Evaluation Process

Metrics. To evaluate the model's performance, we computed the metrics defined below by comparing the golden standard annotations with the output of the automatic system.

The metrics established are P (Precision), R (Recall), and F-Measure.

1. Precision is defined by the ratio of correct answers (True Positives) among the total answers produced (Positives),

$$P = \frac{TP}{TP+FP} \quad (1)$$

where TP - True Positive, a predicted value was positive and the actual value was positive and FP - False Positive, predicted value was positive and the actual value was negative.

2. R - Recall is defined as a ratio of correct answers (True Positives) among the total possible correct answers (True Positives and False Negatives),

$$R = \frac{TP}{TP + FN} \quad (2)$$

where FN - False Negative, a predicted value was negative and the actual value was positive.

3. F1-score is a harmonic mean of precision and recall,

$$F1_{score} = \frac{2*P*R}{P+R} \quad (3)$$

Additional evaluation for NER models. This section explains the evaluation method used to compare the different approaches used for training evolutive models with the vanilla model, i.e., the original pre-trained model.

The utility and difficulty of recognizing some types against some others are different, demonstrating the need for a study at the entity level. Therefore, we go beyond simple token-level performance and evaluate each entity type recognized in the corpus.

We define the accuracy ratio as:

$$ratio = \frac{\#correct - \#incorrect}{\#correct + \#incorrect} \quad (4)$$

where #correct represents the total number of correct predictions and #incorrect, the total number of incorrect predictions.

- A positive ratio means that the model is overall making correct predictions. (|correct| > |incorrect|)
- A ratio of 1 means that all the model predictions are correct. (|incorrect| = 0)
- A negative ratio means that the model is overall making incorrect predictions. (|correct| < |incorrect|)
- A ratio of -1 means that all the model's predictions are incorrect. (|correct| = 0)
- A ratio of 0 means that the model is balanced. (|correct| = |incorrect|)

Each time we review and correct a document, it is considered the gold standard and compares the model's prediction based on the correct outcome. The different ratio values of each document are saved and plotted in an accumulative ratio graph. The accumulative ratio curve is easier to interpret as it visualizes the global trend of the model performance evolution.

4 Results & Discussion

4.1 Dataset

Following the semi-automatic approach, we create a labeled dataset for NER and SRE using the annotation scheme described in section 4.6. The data was annotated with the help of a domain specialist to ensure the annotation guideline we followed was concordant with the user's needs and requirements. The generated corpus consists of 3063 documents, 348503 tokens, and 16780 sentences in total. The count and proportion of each entity and relation tag are given in Table 3. And Table 4.

Table 3. Summary of the number of NER tags. Each line represents the number of entities found in each type of documents.

Named Entities									
Document	**PER**	**ORG**	**GPE**	**POI**	**ADR**	**FAC**	**NORP**	**DATE**	**TIME**
MO	413	194	729	252	107	418	98	1085	656
PDF	994	602	2773	1059	358	341	537	815	1313
GV	539	6	651	225	177	13	114	2292	463
TOTAL	1946	802	4153	5136	642	772	749	4192	2432
Named Entities									
Document	**WEAPON**	**CRIME**	**CELM**	**MONEY**	**WEB**	**EVENT**	**DRUG**	**PROF**	**TOTAL**
MO	113	945	1946	244	74	9	216	354	7975
PDF	226	2244	1496	291	38	4	481	597	14244
GV	0	191	52	0	0	0	103	362	5197
TOTAL	339	3380	3494	535	112	13	800	1313	27416

Table 4. Summary of the number of relation types in the generated corpus.

Relation Label	Number of tuples
PER-PER	318
PER-NORP	127
PER-DATE	422
PER-GPE	747
PER-CRIME	285
PER-PROFESSION	286
PROFESSION-ORG	120
PROFESSION-GPE	89
ORG-GPE	143
DATE-TIME	2120
CRIME-DATE	89
CRIME-TIME	76
CRIME-GPE	103
CRIME-MONEY	156
EVENT-DATE	56
EVENT-GPE	45

In the following sections, we study relevant statistics about the tool from five aspects: annotation time, annotation effort, annotation quality, annotator satisfaction, and incremental model performance.

4.2 Annotation Time & Effort

Table 5. shows the averaged annotation time for one text according to each method. For the semi-automated mode, the annotators took on average 10 seconds per sentence and 24 seconds for the fully manual mode. The annotation time is approximately two times shorter on average than that of Manual, which indicates an improvement linked to the use of our approach.

During the experimentation, we noticed that the annotation time decreased as annotators got more familiar and experienced with the task. However, the total annotation time decreased even more when using pre-annotated documents.

From these observations, we can conclude that the annotation becomes more efficient with the assistance of pre-annotations. This is further confirmed with the next analysis on the number of correction actions performed in both settings.

Table 5. Comparison of annotation time in seconds for the two annotation modes

Method	Averaged Time per sentence (seconds)
Manual	23.61
Semi-Automatic	10.27

Table 6. Comparison of the total number of actions for the two annotation modes

Method	Number of actions per sentence
Manual	15
Semi-automatic	8

Table 6 displays the average number of actions performed in each mode. Again, as claimed above, the total number of corrections is lower on average with pre-annotated documents.

We conclude that the overall human effort decreases with the introduction of pre-annotations.

4.3 Annotation Quality

Manual inspection.
To assess the quality of annotations with and without prediction assistance, we created a gold standard corpus with the help of an expert annotator from scratch on the documents used for the first experiment. The documents were not pre-annotated to avoid bias induced in the review phase. We then evaluate the curations generated by the different annotators on this corpus to evaluate their precision, recall, and F1-score values. The results of this analysis are reported in Table 7.

Table 7. Averaged annotation quality in terms of Precision, Recall and F1-score for the two annotation modes

Method	Precision	Recall	F1-score
Manual	82%	71%	76%
Semi-automatic	80%	85%	82%

These results indicate that annotations are, on average, more consistent with the presence of pre-annotations. Therefore, the overall annotation quality is higher in these conditions as annotators seem to make fewer errors. A possible explanation is that the automatic task performs relatively easy tasks such as detecting dates or times, and the other complicated ones are left for the human. Therefore, their focus is reduced to the essential and complex cases, making the annotator less prone to make errors.

Training Data for the automatic model. To further validate the quality of the semi-automatic process on a larger set of data, we train a final model using the generated corpus during the second experimentation. We split the data into a training set (90% of the total data) and a test set (10% of the total data) and obtain the following results.

Precision: 88% Recall: 86% F1-score: 87%

We calculate the accuracy score, precision, recall, and f1-score for each entity that the model recognizes for each document. The values of these metrics for each entity are summed up and averaged to generate an overall score to evaluate the model on the data. The entity-wise evaluation results are summarized in Table 7.

In order to have a fair evaluation of our model, we use human-annotated datasets as validation of our model. We use the English CoNLL-2003 dataset [29], as the gold standard for model evaluation. This gave us an F-score of 91.7% on this dataset which validates the quality of the generated corpus.

4.4 Annotator Feedback

At the end of the experimentation, annotators were asked to assess their satisfaction with the tool. Most annotators agreed that the user interface functionalities made the process more pleasant. They especially appreciated the automatic boundary snapping functionality during token selection. It was generally observed that annotators were comfortable and rapidly got familiar with the tool. They also reported that less manual look-up was required. Overall ratings of the tool were positive except for some negative comments focusing primarily on difficulties understanding the feedback process in general and details of exactly how the automatic algorithms operated.

4.5 Incremental Models' Performance

This section compares the performance of incremental models to a traditionally trained model and summarizes the results in Table 8. The table shows that the iterative models were able to overcome overfitting and generalize well. The final model has a higher F1-score, but the distance between these systems is not significant. This was also observed during the annotation of the final corpus documents. The auxiliary model's predictions were fairly accurate, and the human annotator added only a few modifications. This achieved our goal of switching from an active annotation mode to a reviewing mode. These findings can be explained by the high regularisation used over the iterative models to prevent them from overfitting.

Table 8. Performance scores of evolutive models and the baseline in terms of Precision, Recall and F1-score

Dataset	Training Set			Test Set		
	Precision	Recall	F1-score	Precision	Recall	F1-score
1by1	89.05%	87.58%	88%	83.38%	82.3%	82.84%
10by10	89.89%	87.47%	88.67%	84.44%	82.56%	83.49%
all	90.38%	87.59%	88.96%	88.44%	86.56%	87.49%

We perform another evaluation to closely analyze the accuracy evolution of the iterative models throughout the experimentation. Figures 3 and 4 show the accumulative number of correct predictions on every document. The training iterations represent the number of documents manually reviewed. We compare the performance of the incremental models on the new entities added in Figure 3. In Figure 4, we compare the performance of the iterative models and the vanilla model that only recognizes a set of global entities. We use the vanilla model as a baseline for our evaluation.

The figures show that both models could learn new entities over the iterations as the curve is increasing steadily until it reaches a stable point. It can also be seen that the training size has a significant impact on model learning. There is a general trend of increasing accuracy the more documents are labeled. This observation was also noticed during the experimentation. Indeed, after annotating over 500 documents, the models were able to output correct predictions and tag these new entities correctly. This significantly improved the annotation time as it reduced the number of corrections required. Overall, both models' performance was similar with a noticeable difference in predicting the tags: CRIME, ADDRESS, DRUG, and PERIOD. For these entities, the 10 by10 model prediction was more accurate. A possible reason could be the noisy steps introduced by the frequent updates on the 1by1 model. Updating the model each time a modification is performed could add a noisy gradient signal as observed when comparing Stochastic Gradient Descent with Mini-Batch Gradient Descent.

Fig. 3. Accumulative ratio of correct predictions on new entities

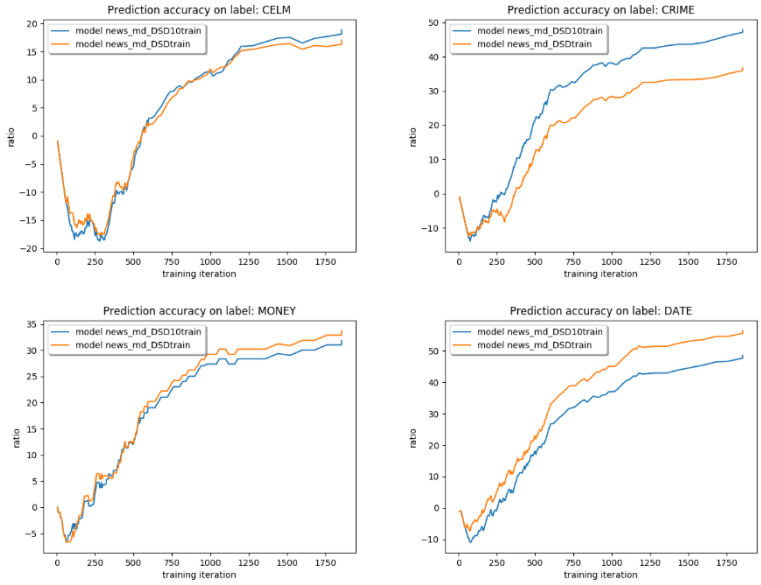
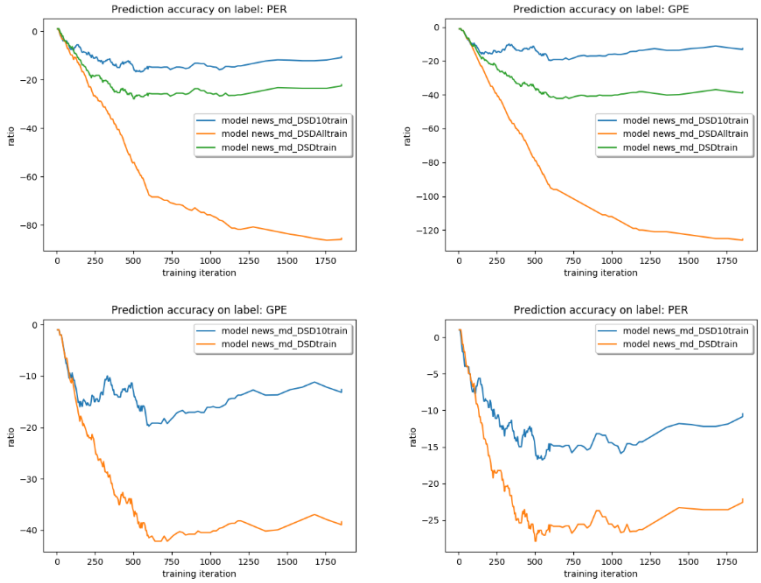


Fig. 4. Accumulative ratio of correct predictions on old entities



It can seem surprising that the vanilla model performed poorly on old entities, i.e. entities trained to detect on a large corpus of documents. This is due to the fact that we changed the definition of these entities slightly by including tokens in the tags that were not considered before. For example, we include the postcode of the city in the GPE definition and also the person title in PER to differentiate between civilians and servicemen. The gap between the vanilla model and the iterative models shows that these models could learn and adapt to the updated entities' definitions. Overall, we notice that the 10 by10 model achieves higher accuracy compared to the vanilla baseline and 1by1 model.

5 Conclusion

This paper presents a semi-automatic annotation tool based on an iterative learning process to reduce human intervention. We evaluate the effectiveness of our method on two NLP tasks that perform Named Entity Recognition and Semantic Relation Extraction for general and criminal entities and relations between them. We adapt the automatic systems to cover a large number of languages and improve their architecture to achieve higher accuracy.

The qualitative and quantitative evaluations of the proposed method indicate that this process reduces human effort in both annotation time and correction actions while preserving the annotation quality. Furthermore, the semi-automatic annotation helped reduce the annotation time due to knowledge propagation and improve the annotation quality by finding inconsistencies and making changes in the annotation guideline early and not until the end. Following this technique, we generate a dataset for French NER and SRE for general and crime fields. We used this data to train multilingual prediction models via zero-shot transfer learning and achieve an F1-score of 87% and 81% for NER and SRE respectively. The results have also revealed that iterative models' final performance was close to the traditional model and that update intervals have a noticeable impact on accuracy as the model re-trained on an interval of 10 documents outperformed the one trained at each document correction.

These findings can have many implications as the underlying method can be applied to other multimedia applications and be of great use for large-scale annotation campaigns.

There are other areas in which we can further evaluate and enhance the performance of the system. In the future, we plan to support multi-user access to the tool with high-level supervision and management of conflicts. Further experimental investigations are needed to evaluate the impact of the incremental learning approach against a traditional semi-automatic annotation at effort measurement and annotation quality levels. Due to time constraints, it was not possible to do these experiments as part of the current work. Additionally, in this paper, we only compared two approaches for the evolutive models. An interesting study could be to try to find the optimal trade-off interval value. This experiment showed that a low training interval value could add a noisy gradient element

and disrupt the training. Meanwhile, choosing a high value will not rapidly convey the knowledge introduced by the manual reviewing. A future study could also investigate the optimal hyperparameter tuning strategy to train incremental models.

References

[1] Kulkarni, Sheshagiri & Nath, S Shashi & Pandian, B.. (2003). Enterprise information portal : a new paradigm in resource discovery.

[2] J. Li, A. Sun, J. Han and C. Li, A Survey on Deep Learning for Named Entity Recognition, in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2020.2981314, 17 March 2020.

[3] Sachin Pawar, Girish K. Palshikar, Pushpak Bhattacharyya, Relation Extraction : A Survey, arXiv:1712.05191 [cs.CL], 14 Dec 2017.

[4] Kanako Komiya, Masaya Suzuki, Tomoya Iwakura, Minoru Sasaki, and Hiroyuki Shinnou. 2018. Comparison of Methods to Annotate Named Entity Corpora. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 17, 4, Article 34 (August 2018), 16 pages. DOI:https://doi.org/10.1145/3218820

[5] M. Y. Akpınar, B. Oral, D. Engin, E. Emekligil, S. Arslan and G. Eryiğit, "A Semi-Automatic Annotation Interface for Named Entity and Relation Annotation on Document Images," 2019 4th International Conference on Computer Science and Engineering (UBMK), 2019, pp. 47-52, doi: 10.1109/UBMK.2019.8907209.

[6] Kuzman Ganchev, Fernando Pereira, Mark Mandel, Steven Carroll, and Peter White. 2007. Semi-automated named entity annotation. In Proceedings of the Linguistic Annotation Workshop (LAW '07). Association for Computational Linguistics, USA, 53–56.

[7] Halike, A.; Abiderexiti, K.; Yibulayin, T. Semi-Automatic Corpus Expansion and Extraction of Uyghur-Named Entities and Relations Based on a Hybrid Method. Information 2020, 11, 31. https://doi.org/10.3390/info11010031

[8] C. Cano, A. Labarga, A. Blanco and L. Peshkin, "Collaborative semi-automatic annotation of the biomedical literature," 2011 11th International Conference on Intelligent Systems Design and Applications, 2011, pp. 1213-1217, doi: 10.1109/ISDA.2011.6121824.

[9] Névéol A, Islamaj Doğan R, Lu Z. Semi-automatic semantic annotation of PubMed queries: a study on quality, efficiency, satisfaction. J Biomed Inform. 2011;44(2):310-318. doi:10.1016/j.jbi.2010.11.001

[10] Liu, W., Dumais, S., Sun, Y., Zhang, H., Czerwinski, M., & Field, B.A. (2001). Semi-Automatic Image Annotation. INTERACT.

[11] Simone Bianco, Gianluigi Ciocca, Paolo Napoletano, and Raimondo Schettini. 2015. An interactive tool for manual, semi-automatic and automatic video annotation. Comput. Vis. Image Underst. 131, C (February 2015), 88–99. DOI:https://doi.org/10.1016/j.cviu.2014.06.015

[12] Laclavik, M. et al. "Ontea: Platform for Pattern Based Automated Semantic Annotation." *Comput. Informatics* 28 (2009): 555-579.

[13] Teixeira, Jorge. (2011). Automatic Generation of a Training Set for NER on Portuguese journalistic text.

[14] Hoxha, Klesti & Baxhaku, Artur. (2018). An Automatically Generated Annotated Corpus for Albanian Named Entity Recognition. Cybernetics and Information Technologies. 18. 10.2478/cait-2018-0009.

[15] Canito, Alda et al. "Automatic Document Annotation with Data Mining Algorithms." *WorldCIST* (2019).

[16] Menezes, Daniel Specht et al. "Building a Massive Corpus for Named Entity Recognition Using Free Open Data Sources." 2019 8th Brazilian Conference on Intelligent Systems (BRACIS) (2019): 6-11.

[17] Alex, Beatrice et al. "Agile Corpus Annotation in Practice: An Overview of Manual and Automatic Annotation of CVs." Linguistic Annotation Workshop (2010).

[18] P. Stenetorp, S. Pyysalo, G. Topic, T. Ohta, S. Ananiadou, and J. Tsujii. Brat: A web-based tool for nlp-' assisted text annotation. In Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL '12, pages 102–107, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics

[19] H. Cunningham, V. Tablan, A. Roberts, and K. Bontcheva. Getting more out of biomedical documents with gate's full lifecycle open source text analytics. PLOS Computational Biology, 9(2):1–16, 02 2013

[20] French, Robert. (1999). Catastrophic forgetting in connectionist networks. Trends in cognitive sciences. 3. 128-135. 10.1016/S1364-6613(99)01294-2.

[21] Goodfellow, I. et al. "An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks." *CoRR* abs/1312.6211 (2014): n. pag.

[22] Li, Zhizhong & Hoiem, Derek. (2016). Learning Without Forgetting. 9908. 614-629. 10.1007/978-3-319-46493-0_37.

[23] Devlin, J. et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT* (2019).

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[25] Hugging Face – The AI community building the future, https://huggingface.co/, last recovered 17/Jun/2021.

[26] Batista, David & Martins, Bruno & Silva, Mário. (2015). Semi-Supervised Bootstrapping of Relationship Extractors with Distributional Semantics. 10.18653/v1/D15-1056.

[27] Porter, M.. "Snowball: A language for stemming algorithms." (2001).

[28] Reimers, Nils & Gurevych, Iryna. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 3973-3983. 10.18653/v1/D19-1410.

[29] Sang, Erik & Meulder, Fien. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. Proceeding of the Computational Natural Language Learning (CoNLL). 10.3115/1119176.1119195.