Iterative Learning for Semi-automatic Annotation Using User Feedback

Meryem Guemimi, Daniel Camâra, and Ray Genoe

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 french 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.

Index Terms—Semi-Automatic Annotation, Natural Language Processing, Named Entity Recognition, Semantic Relation Extraction, 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 multiple documents per year such as criminal reports, signaling from citizens and companies. Only in terms of formal complaints the CSCI receives approximately 1.8 Million each year. These documents, sent by heterogeneous and voluntary sources, come mostly in unstructured form, making it impossible to impose or even control the reports format. 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 addressed in this paper. Today we may find different frameworks proposing generic pre-trained models. However, the lack of domain-specific knowledge makes them unsuitable for certain fields. Law enforcement is not an exception. The vocabulary of the analyzed documents and information of interest vary significantly from those proposed by the regular frameworks. In this situation, transfer learning or even the full retraining of available models may be required which implies the annotation of a substantial number of documents.

This paper describes our efforts to build a system that simplifies and speeds up annotation. We propose a semi-automatic tool for textual information annotation that combines the efficiency of automatic annotation and the accuracy of manual annotation. We investigate the validity of the proposed method on Named Entity Relation extraction (NER) [2] and Semantic Relations Extraction (SRE) [3]. Many other tasks could be used within the annotator framework, but they are not the focus of this work. We use state-of-the-art pre-trained models capable of extracting general named entities and relations between them. As the user provides domain-specific text and corrects model predictions by modifying or adding missing elements, a background training process launches. A transfer learning strategy with fine-tuning is utilized to enable injecting user knowledge into models. After several iterations, the model's accuracy becomes high enough that we switch from an annotation mode to a reviewing mode, which reduces the amount of manual labor and level of expertise required to annotate domain-specific texts.

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 analyses the results and Section 5 concludes the paper.

- *M. Guemimi and D. Camâra are with the Center for Data Science, Judiciary Pôle of the French Gendarmerie, Pontoise, France. E-mail: {meryem.guemimi, daniel.camara}@gendarmerie.interieur.gouv.fr*
- *R. Genoe is with the Center for Cybersecurity and Cybercrime Investigation University College Dublin, Dublin, Ireland. E-mail: ray.genoe@ucd.ie*

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. To cope with the high demand for annotated training corpora, researchers have been exploring techniques to derive better annotation systems that minimize human effort. Different techniques have been proposed to partially or fully automate the annotation process. In this section, we present a selection of these studies. No attempt is made to be exhaustive, as the goal is to compare and contrast these with our efforts.

2.1 Semi-automatic Approaches

Semi-automatic text annotation combines automatic system predictions with human corrections by asking a human annotator to revise an automatically pre-tagged document instead of doing 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 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. 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.

Semi-automatic approaches are generally found helpful by most annotators; however, they still require human intervention and are not efficient when applied to specific domains far from which the automatic model was trained [4]. 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 indicates 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 on the object detection module. Results demonstrate that the system reduces the average ratio of human intervention.

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 Natural Language Processing (NLP) models based on user feedback. 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. Even though the above mentioned 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.

2.3 Fully Automatic Approaches

While semi-automated 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. The success rate of the technique is highly dependent on the definition of the patterns, but it 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 methods 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

3.1 Proposed Strategy

This paper presents an NLP annotator platform that automatically identifies and tags entities and relations between them in plain text. The human annotator then corrects the model prediction instead of annotating the text from scratch. The strategy is to iteratively update and refine the inference models via fine-tuning, until the whole corpus is annotated. The goal is to change the task from a manual annotation to a manual reviewing by using corrections introduced, 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. This helps increase models precision on known tags while learning new classes identified by users. If annotators identify during the annotation process, a new class that is of interest, model's architecture is adapted accordingly. After a few examples, as models learns, the new class will naturally start appearing on the next pre-annotated documents. Additionally, this method allows to revise or correct possible flaws in the annotation guidelines early rather than at the end as in traditional linear annotation methods [17]. Potential sources of error occurring throughout the annotation process (e.g., inconsistent annotation, ambiguous guidelines) are recognized in a decrease in incremental models performance. The early detection of these flaws allows to reduce the correction cost and increase annotation quality.

3.2 Tool

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 required functionalities included automatic tagging of entities and relations, the addition of new classes to the annotation scheme and model finetuning. Additionally, due to the sensitivity of the dataset used during the experimentation, we decided to design a custom tool.

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 platform supports different input and output format such as spaCy's JSON, CoNLL-IOB, and CoNLL-BILUO.

The interface is designed to be intuitive and user-friendly. 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 Figure 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 Process

A typical user scenario goes as follows. First, the user prepares a dataset in a supported format and uploads it to the tool. As shown in Figure 2, the platform automatically sends an annotation request to the server and displays the results in the GUI. The user can review and correct the pre-annotated document and then move to the next one. In the background, a training request is sent to the server, with the last text manually corrected. The automatic model is finetuned based on user feedback. Once complete, the server stores the updated models and uses them for the following inference round. This process repeats until the whole corpus is annotated. The server treats inference and training requests asynchronously, which avoids adding any possible overhead due to model retraining, making the training process transparent to users during the daily use of the system.

3.4 Training 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 domain 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. However, this approach may suffer from a performance degradation on old tasks, also known as catastrophic forgetting [20], [21]. When trained on one task, then trained on a second task, models may "forget" how to perform on the first one. This is especially observed for neural network-based systems. Different fine-tuning adaptation techniques have been studied to reduce this effect and train models on new tasks while preserving their original capabilities [22]. Our case is simple since the old task, i.e., original set of classes to recognize, can be seen as a subset of the new one that introduces new classes. For this reason, we use a rehearsal regime that mixes new annotations alongside old ones to recall items representative of the previous task.

The goal is to perform model expansion while still using the original network's information. Ideally, the new tasks could be learned while sharing parameters from the old ones without retraining from scratch. For this reason, we

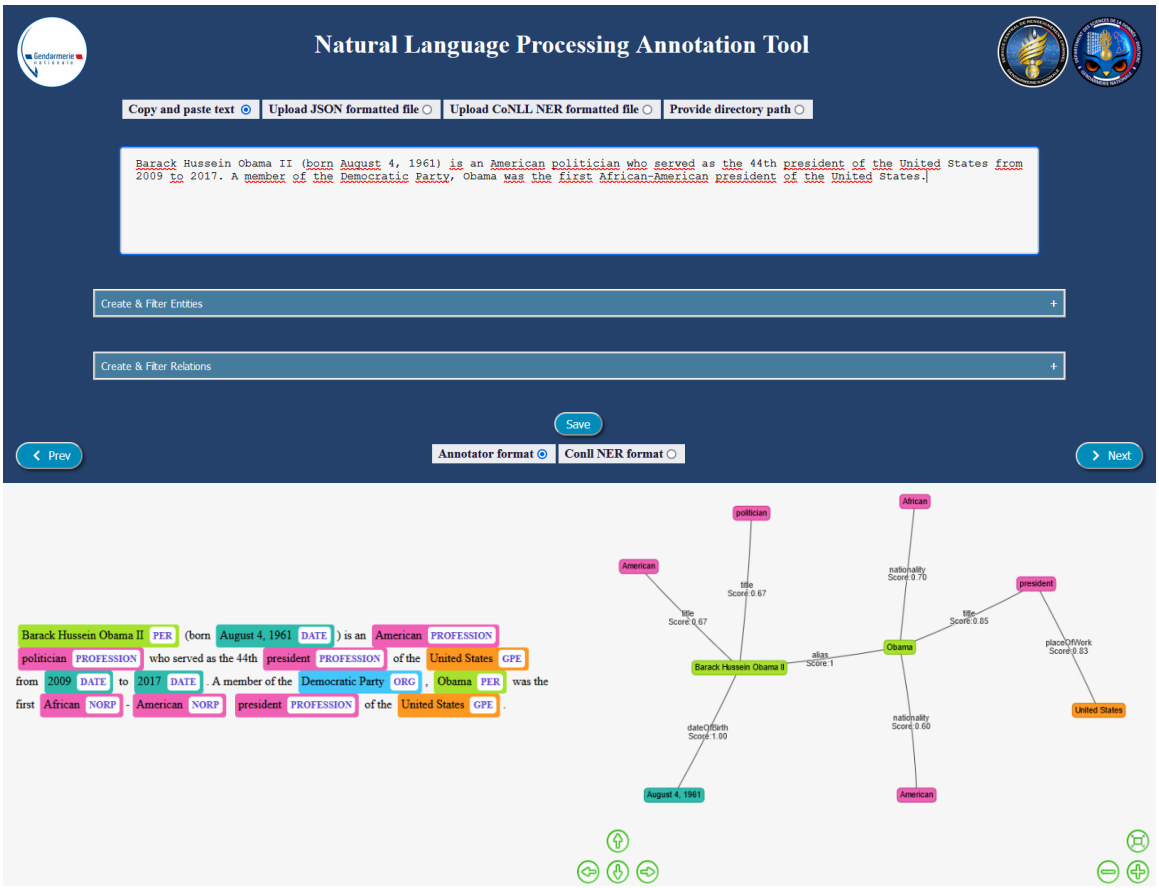


Fig. 1: Natural Language Processing Annotator User Interface. A sample pre-annotated text is displayed. The raw text is visible in the top input area. Entities are highlighted on the left hand side and a relational graph connecting the linked entities is visible on the right hand side. The user can, from this view, edit the annotation and save the result in the exportable format at any time.

preserve the original weights of the previous architecture and add new task-specific nodes with randomly initialized parameters fine-tuned at a later stage. The fine-tuning operates at the classifier nodes as low layers' weights are frozen. However, when the corpus size is large, the model retraining becomes costly. This limitation could halt the intended gains of the proposed approach, leading to less favorable solutions such as full manual annotation. For this reason, to tackle the possible catastrophic forgetting, at each training call, we retrain the model on the latest corrected texts mixed with a random sample of previously reviewed documents instead of using on the whole corpus. As the model requires a few examples to learn a completely new class, we use a heuristic for training data construction that ensures an upsampling of the latest documents. 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 enhance fine-tuning on new classes and rapidly converge the new weights. On average, it is expected that the training set contains a fair distribution of most of the classes recognized previously by the model along with the newly introduced ones.

In what follows, j denotes the current training round and N the number of antecedent batches used in the sampling. To construct the training set $data_j$ at round j, we take a random sample of elements from each batch without replacement, as follows:

$$data_j = \bigcup_{i=0, i\leq j}^{N} sample(\lceil \alpha_{i,N} * |B_{j-i}| \rceil, B_{j-i}) \quad (1)$$

where B_i refers to the batch of documents received at round $i \in [\![0, j]\!]$ and $\alpha_i \in [0, 1]$ represents the sample size in percentage. The speed at which the older annotations are dampened is a function of the value of α. We define the sample size α_i associated to batch B_{j-i} as:

$$\alpha_{i,N} = \begin{cases} 1 - \dfrac{i}{N} & (2a) \\ \dfrac{1}{N} \quad \text{if SMA} & (2b) \\ a^i, \text{ with } a \leq 1 \quad \text{if EWMA} & (2c) \end{cases}$$

$\alpha = 1$ for the most recent data, indicating that we use all the examples received from this batch. At the next round, this ratio drops to $\frac{N-1}{N}$ and decreases until round $j + N$ from which we no longer include elements of this batch on the training set. This guarantees an upsampling of the last received documents to speed up the learning of new classes and jointly improve the model performance on old classes.

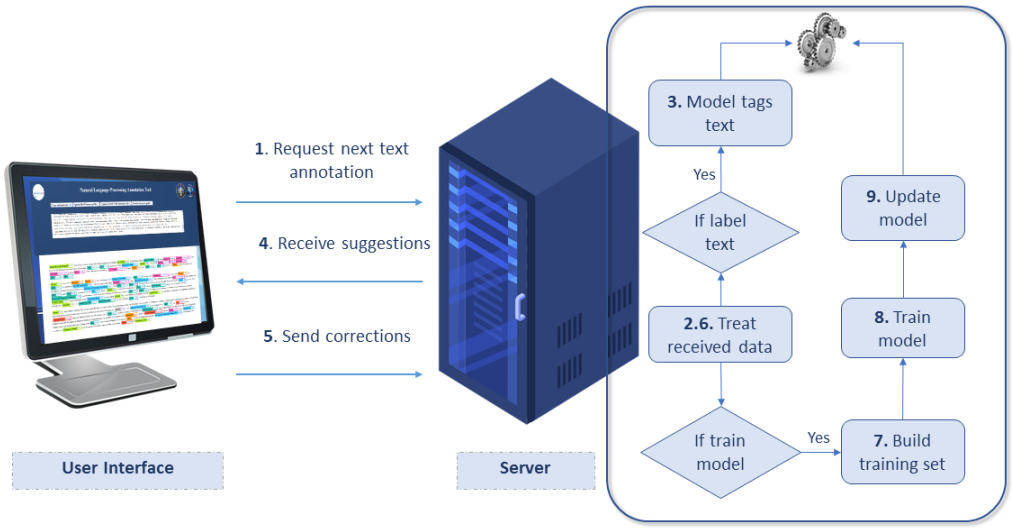


Fig. 2: System architecture showing the annotation and incremental training processes.

We explored alternative regimes to include previous items such as a Simple equally weighted Moving Average (Equation 2b) or an Exponentially Weighted Moving Average (Equation 2c). Experiments with these different sampling schemes have shown that our heuristic combines the benefits of both techniques. As with the running average, this method preserves the overall statistical distribution of the different classes in the training set. It produces a fair dataset, i.e., close to the base population class distribution. Similar to the exponential smoothing approach, it allows speeding up the learning process of new classes by giving greater weights to more recent data that potentially is of higher accuracy and may contain new tags of entities.

The iterative training approach may be prone to overfitting as the model is re-trained multiple times on repetitive data. For this reason, we use a decaying dropout rate to tackle the small data size at the beginning of the annotation. However, overfitting 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 4.5.

3.5 Framework selected

Many frameworks can be used to assist the annotation process, as long as they support a continuously learning strategy. 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.

3.5.1 Hugging Face

To perform NER, we use a BERT-based Transformer model. BERT [23] is a pre-trained transformer network [24], which sets for various NLP tasks new state-of-the-art results. It was trained on a large corpus data in a self-supervised fashion using a masked language modeling objective. This enables the model to learn an internal representation of the languages in the training set that can then be used to extract features useful for downstream tasks. We fine-tune 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.

3.5.2 BREDS

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 expands a set of initial seeds by automatically generating extraction patterns. Bootstrapping approaches don't 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. We improved the original BREDS pipeline to expand the extraction to non-verb mediated relations and replaced the word embedder with a 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 its validity, we perform two different experiments.

3.6.1 Experiment 1

Seven 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 annotations quality.

3.6.2 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.

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.

3.6.3 Dataset

The dataset used consists of real, non-confidential, and anonymized documents collected from the internal database of CSCI. The documents were randomly extracted to avoid bias and have a large representation of the knowledge database. It consists of several texts, including aggregated crime reports, complaints, with a focus on the Modus Operandi (MO) free text field, and procès-verbaux (GV for Garde à Vue).

3.6.4 Annotation Scheme

The annotation scheme was first developed after inspecting, with domain experts, the entities and relations of interest. It was further enriched after a first manual annotation campaign. Special cases encountered at this stage, helped adjust the defined guidelines. For instance, some MO texts did 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).

The final annotation scheme for Named Entities is presented in Table 1 and Semantic Relations in Table 2.

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, cdg airport, 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	words or expressions referring to an infraction.
WEB	web activities (website, social network, cyber activity, etc.).
EVENT	festivals, sports events, etc.

3.6.5 Evaluation Process

3.6.5.1 Metrics: To evaluate the model's performance, we compute the P (*Precision*), R (*Recall*) and *F1-score* metrics by comparing the golden standard annotations with the output of the automatic system.

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 (Birth place), ADDRESS (address), PROFESSION (job), CRIME (victim, assailant).
PROFESSION-OBJ	ORG (organisation 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).

1) *Precision* is defined as the ratio of correct answers among the total answers produced

$$P = \frac{TP}{TP + FP} \tag{3}$$

where $TP - TruePositive$, is the number of correctly labelled positive samples and $FP - FalsePositive$, the number of negative samples incorrectly labelled as positive.

2) *R - Recall* is defined as the ratio of correct answers among the total possible correct answers

$$R = \frac{TP}{TP + FN} \tag{4}$$

where *FN - False Negative*, is the number of positive samples incorrectly labelled as negative.

3) *F1-score* is the harmonic mean of precision and recall

$$F1_{score} = \frac{2 * P * R}{P + R} \tag{5}$$

3.6.5.2 Incremental NER models: An additional evaluation method is used to compare the different approaches used for training evolutive models against the vanilla model, i.e., the original pre-trained model.

The utility and difficulty of recognizing some types varies. Therefore, we go beyond simple token-level performance and evaluate each entity type detected in the corpus. We define the accuracy ratio as

$$ratio = \frac{Correct - Incorrect}{Correct + Incorrect} \tag{6}$$

where $Correct$ represents the total number of correct predictions (TP) and $Incorrect$, the total number of incorrect predictions ($FP + FN$).

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

The ratio is computed every time a new document is reviewed and corrected by a human annotator. The values are saved and plotted in an accumulative ratio graph. The accuracy ratio fluctuates around the zero line from one document to another as the distribution of entities varies significantly. For this reason, we compute the aggregated amount of correct predictions and display it in a curve to better visualize the global trend of the model performance evolution during the annotation campaign. A steady increase of this curve indicates a continuous positive ratio and vice versa.

4 Results & Discussion

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.

4.1 Dataset

Following the semi-automatic approach, we create a labeled dataset in French for NER and SRE using the annotation scheme described in section 3.6.4. The data was annotated with the help of a domain specialist to ensure the annotation guideline followed was concordant with the user's needs and requirements. The generated corpus consists of 3063 documents, 348503 tokens, 16780 sentences, 34831 entities and 5310 relation tuples in total. The distribution of each entity and relation class are given in Figure 3 and Table 3 respectively.

TABLE 3: Distribution of relation types in the generated corpus.

Relation Type	Total number of tuples
PER-PER	318
PER-NORP	127
PER-DATE	422
PER-GPE	747
PER-CRIME	285
PER-PROFESSION	286
PER-ADDRESS	128
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

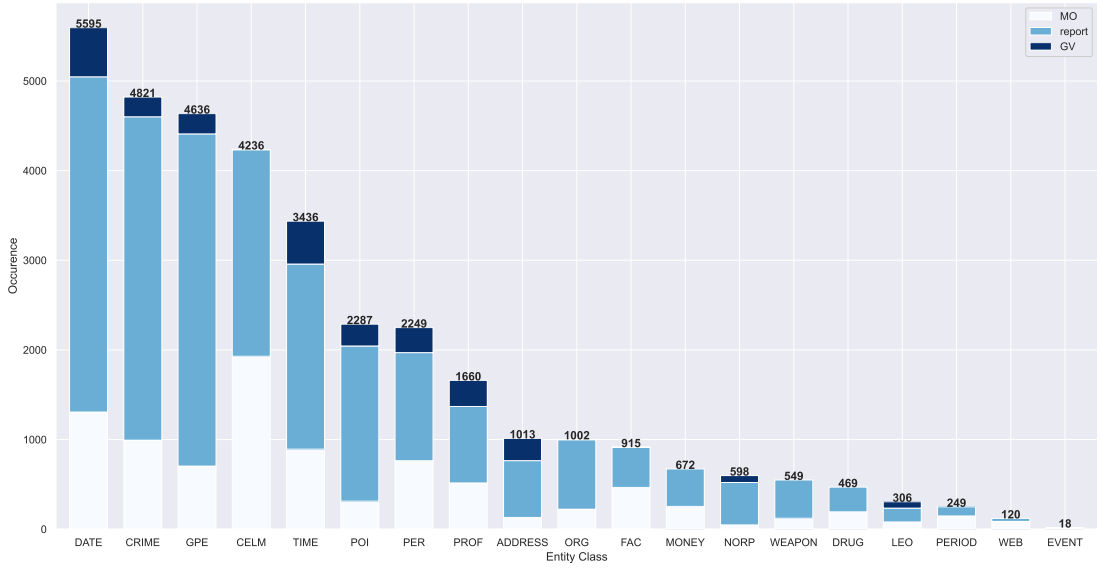


Fig. 3: Distribution of entity types in the generated corpus per document type (Section 3.6.3).

4.2 Annotation Time & Effort

Table 4 shows the averaged annotation time and number of actions according to each method. The annotation time is approximately two times shorter 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 faster with the curation assistance. This is further confirmed with the analysis on the number of correction actions performed in both settings.

TABLE 4: Comparison of Annotation Time and Number of Actions for the Two Annotation Modes.

Method	Averaged Time per sentence (seconds)	Number of actions per sentence
Manual	23.61	15
Semi-Automatic	10.27	8

4.3 Annotation Quality

4.3.1 *Manual Inspection*

We evaluate the curations generated by the different annotators with respect to a golden standard manually generated by an expert annotator in terms of Precision, Recall and F1-score. The reference corpus was annotated on the documents used for the first experiment, from scratch to avoid bias induced in the review phase. The results, reported in Table 5, 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.

4.3.2 *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 a Precision of 88%, a Recall of 86% and an F1-score of 87% for the NER task.

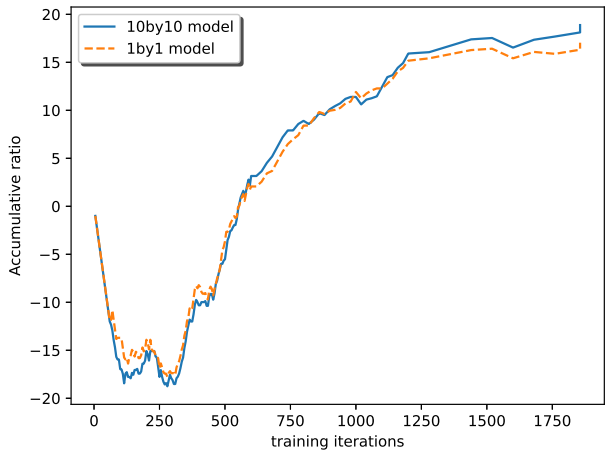
TABLE 5: Averaged Annotation Quality in Terms of P, R and F1 for the Two Annotation Modes.

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

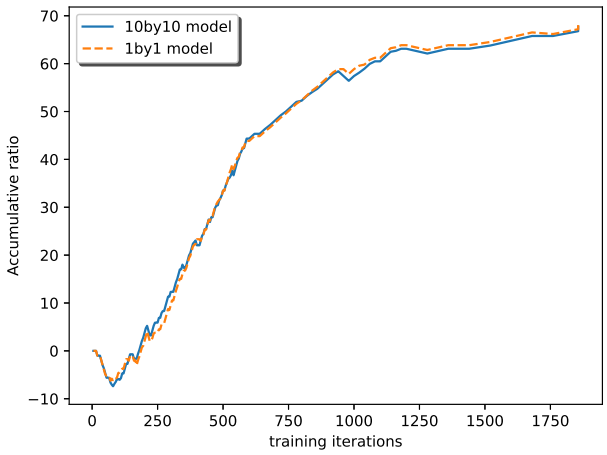
In order to have a fair evaluation of our model, we validate the model performance on a fully human-annotated dataset. We use the CoNLL-2003 dataset [29], as gold standard and achieve an F-score of 91.7% which validates the quality of the generated corpus.

4.4 Annotator Feedback

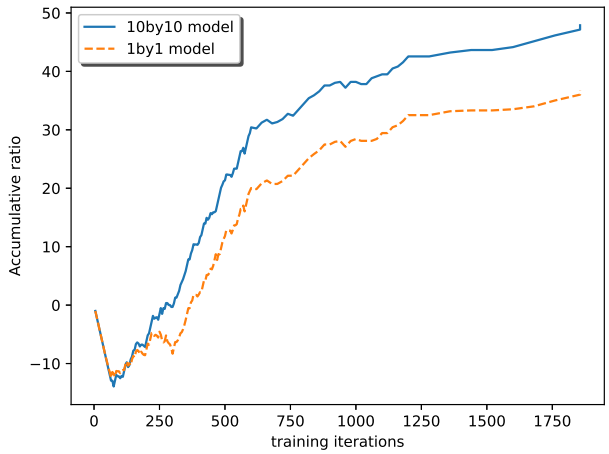
At the end of the experimentation, annotators were asked to assess their satisfaction with the tool. Most annotators agreed that the user interface functionalities made the process more pleasant. They especially appreciated the



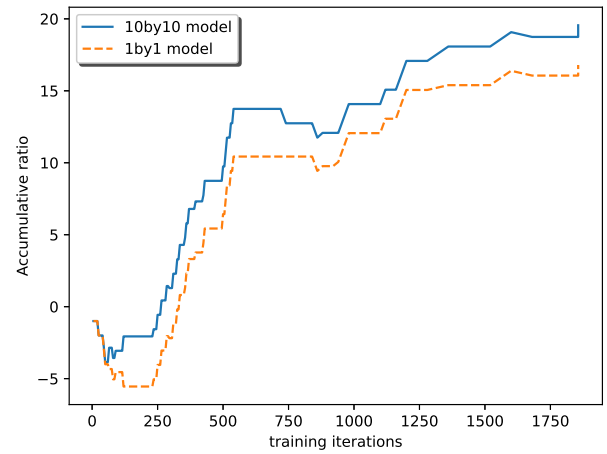
(a) accumulative ratio evolution for class CELM



(b) accumulative ratio evolution for class TIME



(c) accumulative ratio evolution for class CRIME



(d) accumulative ratio evolution for class DRUG

Fig. 4: Comparison of accumulative ratio curve on new entities for incremental models.

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

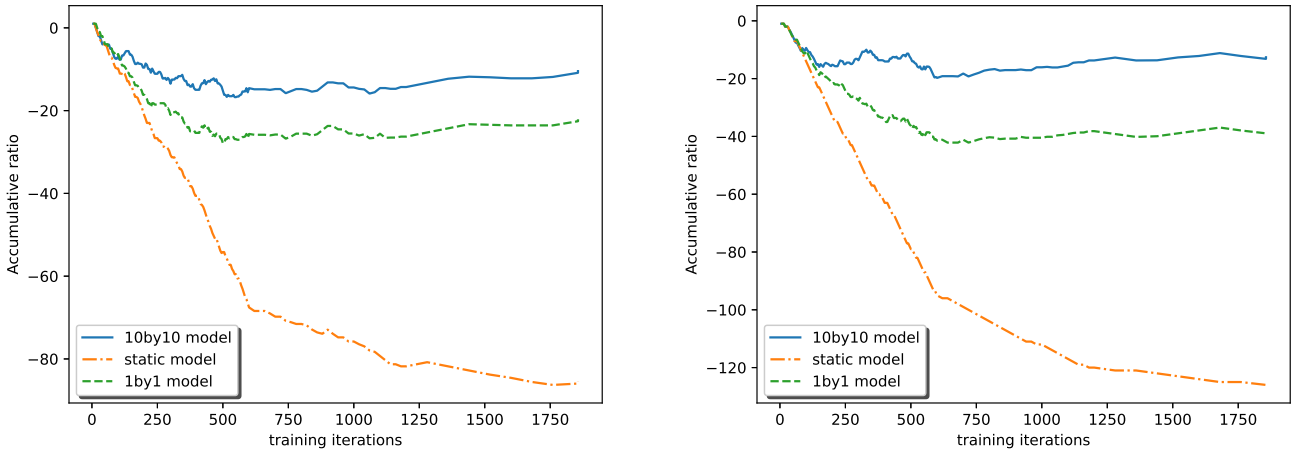
Table 6 compares the performance of incremental models to a traditionally trained model. The results show that the iterative models were able to overcome overfitting and generalize well. The final model has a higher F1-score, but the distance between these systems is not significant. This was also observed 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.

We perform another evaluation to closely analyze the accuracy evolution of the iterative models throughout the experimentation. Figures 4& 5 show the accumulative accuracy ratio evolution for some entity classes. 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 4. The graphs show that both models could learn new entities over the iterations as the curve is increasing steadily until it reaches a stable phase. 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

TABLE 6: Models Performance Scores in Terms of Precision, Recall and F1-score

Model	Training Set			Test Set		
	Precision	Recall	F1	Precision	Recall	F1
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 (baseline)	90.38%	87.59%	88.96%	88.44%	86.56%	87.49%



(a) accumulative ratio evolution for class PER

(b) accumulative ratio evolution for class GPE

Fig. 5: Comparison of accumulative ratio curve on old entities for static and incremental models.

entities correctly. This significantly improved the annotation time as it reduced the number of corrections required. The turning point after which the curve starts strictly increasing varies among classes. For classes with a simple common pattern such as DATE and TIME (4b), the turning point was around 70 documents. Whereas we needed to annotate more than 250 documents to reach this point for more challenging entities such as CELM (4a). Overall, both models' performance was similar as seen for the above entities. However, there was a noticeable difference in predicting the tags: ADDRESS, CRIME (4c), DRUG (4d), NORP, PROFESSION and PERIOD. For these entities, the 10by10 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 algorithms. However, this observation was not true for classes with low occurrences such as EVENT, PERIOD, WEB (Figure 3). This is not surprising as the amount of examples accumulated over 1 or 10 annotations for these classes was not much different due to the class rarity in the used corpus.

In Figure 5, we use the vanilla/static model as a baseline and compare it to the iterative models on only the old set of entities recognized by the original model. 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. 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 10by10 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 NER and SRE for general and criminal entities and relations between them. The proposed method helped reduce the annotation time and number of actions performed by more than 50%, compared to manual curation, and improve the corpus quality. Using the generated dataset, with new annotated classes, we achieve a final F1-score of 87% and 81% for NER and SRE tasks 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. These findings can have many implications as the underlying method can be implemented for 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. Further exper-

imental investigations are needed to assess the impact of incremental learning approaches against traditional semi-automatic methods at annotation quality levels. Due to time constraints, it was not possible to do these experiments as part of the current work. Another interesting study is to find the optimal trade-off interval value for incremental models. This paper showed that a low training frequency could add a noisy gradient element and disrupt the training. Meanwhile, choosing a high value would not rapidly convey the knowledge introduced by the manual reviewing. Additional work has to investigate the optimal hyper parameter tuning strategy to train evolutive models.

ACKNOWLEDGMENTS

This research has been funded with support from the European Commission under the H2020-EU.3.7.1 and H2020-EU.3.7.8 project with Grant Agreement 833276. This publication reflects the views only of the authors, and the European Commission cannot be held responsible for any use which may be made of the information contained therein.

REFERENCES

[1] S. Kulkarni, S. S. Nath, and B. Pandian, "Enterprise information portal : a new paradigm in resource discovery," 01 2003.
[2] J. li, A. Sun, R. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, pp. 1–1, 03 2020.
[3] S. Pawar, G. Palshikar, and P. Bhattacharyya, "Relation extraction : A survey," 12 2017.
[4] K. Komiya, M. Suzuki, T. Iwakura, M. Sasaki, and H. Shinnou, "Comparison of methods to annotate named entity corpora," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 17, no. 4, Jul. 2018. [Online]. Available: 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," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 2019, pp. 47–52.
[6] K. Ganchev, F. Pereira, M. Mandel, S. Carroll, and P. White, "Semi-automated named entity annotation," in *Proceedings of the Linguistic Annotation Workshop*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 53–56. [Online]. Available: https://aclanthology.org/W07-1509
[7] Halike, K. Abiderexiti, and Yibulayin, "Semi-automatic corpus expansion and extraction of uyghur-named entities and relations based on a hybrid method," *Information*, vol. 11, p. 31, 01 2020.
[8] C. Cano, A. Labarga, A. Blanco, and L. Peshkin, "Collaborative semi-automatic annotation of the biomedical literature," in *2011 11th International Conference on Intelligent Systems Design and Applications*, 2011, pp. 1213–1217.
[9] A. Névéol, R. Dogan, and Z. lu, "Semi-automatic semantic annotation of pubmed queries: A study on quality, efficiency, satisfaction," *Journal of biomedical informatics*, vol. 44, pp. 310–8, 11 2010.
[10] W. Liu, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski, and B. A. Field, "Semi-automatic image annotation," in *INTERACT*, 2001.
[11] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini, "An interactive tool for manual, semi-automatic and automatic video annotation," *Computer Vision and Image Understanding*, vol. 131, pp. 88–99, 02 2015.
[12] M. Laclavik, L. Hluchý, M. Šeleng, and M. Ciglan, "Ontea: Platform for pattern based automated semantic annotation." *Computing and Informatics*, vol. 28, pp. 555–579, 01 2009.
[13] J. Teixeira, "Automatic generation of a training set for ner on portuguese journalistic text," 07 2011.
[14] K. Hoxha and A. Baxhaku, "An automatically generated annotated corpus for albanian named entity recognition," *Cybernetics and Information Technologies*, vol. 18, 03 2018.
[15] A. Canito, G. Marreiros, and J. Corchado Rodríguez, *Automatic Document Annotation with Data Mining Algorithms*, 04 2019, pp. 68–76.
[16] D. S. Menezes, P. Savarese, and R. L. Milidiú, "Building a massive corpus for named entity recognition using free open data sources," *CoRR*, vol. abs/1908.05758, 2019. [Online]. Available: http://arxiv.org/abs/1908.05758
[17] B. Alex, C. Grover, and R. Shen, "Agile corpus annotation in practice: An overview of manual and automatic annotation of cvs," pp. 29–37, 08 2010.
[18] P. Stenetorp, S. Pyysalo, G. Topić, 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*. Avignon, France: Association for Computational Linguistics, Apr. 2012, pp. 102–107. [Online]. Available: https://aclanthology.org/E12-2021
[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*, vol. 9, 2013.
[20] R. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, pp. 128–135, 05 1999.
[21] I. Goodfellow, M. Mirza, X. Da, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgeting in gradient-based neural networks," 12 2013.
[22] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 2935–2947, 2018.
[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.
[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
[25] Hugging Face, "The ai community building the future," https://huggingface.co/, 2021, accessed: 2021-09-21.
[26] D. Batista, B. Martins, and M. Silva, "Semi-supervised bootstrapping of relationship extractors with distributional semantics," 09 2015.
[27] M. Porter, "Snowball: A language for stemming algorithms," 2001.
[28] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 01 2019, pp. 3973–3983.
[29] E. Sang and F. Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *Proceeding of the Computational Natural Language Learning (CoNLL)*, 07 2003.