Towards a (Semi-)Automatic Urban Planning Rule Identification in the French Language

Maksim Koptelov, Margaux Holveck, Bruno Crémilleux, Justine Reynaud, Mathieu Roche, Maguelonne Teisseire

Towards a (Semi-)Automatic Urban Planning Rule Identification in the French Language

Maksim Koptelov
UNICAEN, ENSICAEN, UMR GREYC,
Caen, France
INRAE, UMR TETIS,
Montpellier, France
maksim.koptelov@unicaen.fr

Margaux Holveck
ICube, Université de Strasbourg,
Illkirch, France
m.holveck@unistra.fr

Bruno Cremilleux
UNICAEN, ENSICAEN, UMR GREYC,
Caen, France
bruno.cremilleux@unicaen.fr

Justine Reynaud
UNICAEN, ENSICAEN, UMR GREYC,
Caen, France
justine.reynaud@unicaen.fr

Mathieu Roche
CIRAD,
UMR TETIS, Univ.n Montpellier,
AgroParisTech, CIRAD, CNRS, INRAE,
Montpellier, France
mathieu.roche@cirad.fr

Maguelonne Teisseire
INRAE,
UMR TETIS, Univ. Montpellier,
AgroParisTech, CIRAD, CNRS, INRAE,
Montpellier, France
maguelonne.teisseire@inrae.fr

***Abstract*—One of the objectives of the Hérelles project is to find new mechanisms to facilitate the labeling (or semantization) of clusters from time series of satellite images. To achieve this, a proposed solution is to associate textual elements of interest with satellite data. The first step in this process consists of an automatic extraction of the information in the form of rules from urban planning documents composed in the French language. To address this challenge, we propose a method which is based on the multi-label classification of textual segments. It includes a special format for representing segments, in which each segment has a title and a subtitle. In addition, we propose a cascade approach aiming to deal with hierarchy of class labels. Finally, we develop several text augmentation techniques for the texts in French, which are able to improve the prediction results. We demonstrate experimentally that the resulting framework correctly classifies each type of segment with more than 90% of accuracy.**

***Index Terms*—Natural Language Processing, supervised learning, data augmentation**

I. Introduction

Land artificialization is a serious problem of modern society. It is considered as one of the principal factors eroding biodiversity, also a net loss of resources for agriculture and forestial or natural areas [1]. In addition, land artificialization increases the risks of natural disasters such as floods and wildfires, which are very costly to the society [2]. The impacts of land artificialization can be significantly reduced if there is a better control over the process. The studies of land artificialization and natural risk management are aimed to address this problem. Our project, Hérelles[1], is a step forward towards improving it.

The first step in the project concerns extraction of rules from urban planning documents related to research sites of our interest. By a *rule* we understand a formal regulation which can be transformed to a constraint in the form of *"if ... then ..."*. For example, the sentence *"If a piece of land can be built on then there must be a road"* is a rule in our interpretation. Our documents are written in French and they contain regulations such as authorizations, obligations and prohibitions regarding land use and development. These rules have a legal value and are enforceable by law. Their application should therefore be observable on the time series of satellite images. To extract such rules in an automatic manner we develop a pipeline based on machine learning.

Most of the state-of-the-art on rule identification exploit supervised learning setting. Generally, data in French are less available, especially annotated, and in the domain of our study in particular. The latter is not available at all, at least in an open access. For this reason we constructed our own corpus by manually annotating the rules and defining a format for their representation [3]. Our data are labeled using 4 different classes which have a hierarchical structure. To perform their classification, we develop a specific framework, which is based on multiple classifiers. To improve the results of one of the classifiers, we perform data augmentation [4]. Finally, to validate our framework we perform a set of experiments using traditional NLP (Natural Language Processing) methods and a state-of-the-art model based on deep learning. The results demonstrate that our framework is able to identify rules of different categories in urban planning documents in the French language with a high accuracy.

In this work, we study the possibility of extracting constraints from French urban planning documents using modern state-of-the-art methods. The contribution of this paper is three-fold. First, we develop an original pipeline to address a specific problem of rule identification in the context of urban planning and natural risk management. To our knowledge, this

[1]https://herelles-anr-project.cnrs.fr

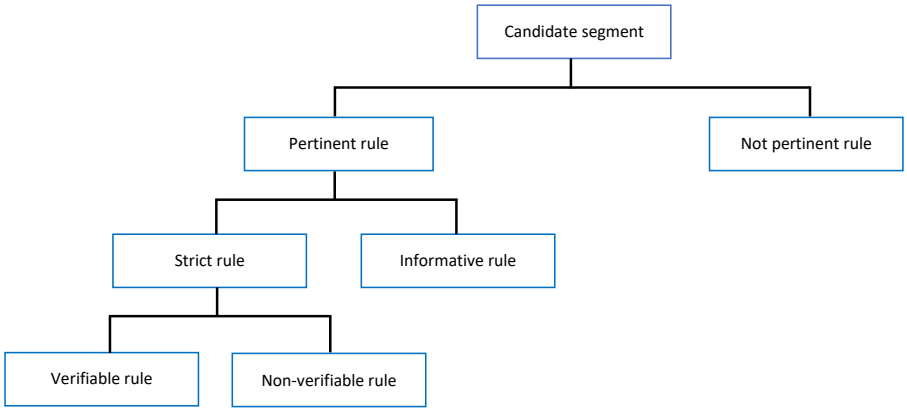

Fig. 1. Hierarchical representation of segments containing rules (Pertinent class) and not (Not pertinent)

is the first work of this type in the aforementioned domains. Second, we propose a cascade approach for multi-label classification of hierarchy of classes. Finally, we designed several text augmentation methods for the French language, which are able to improve the results of text classification. The latter is still rare for the text in French, especially on a topic out of biological or financial domains.

The rest of the paper is organized as follows. Section II provides basic definitions and presents context of the problem. Section III discusses related work on rule identification and text augmentation. Section IV presents our framework for rule identification. Section V describes the data used for experiments, the experimental setup and presents the results. Finally, Section VI concludes and outlines future work.

II. Background

The main research topics of the Hérelles project are the effects of urbanization and natural risk management. The principal research site of the project concerns Montpellier Méditerranée Metropolis (3M) in France, a rapidly evolving area exposed to natural risks.

The final goal of Hérelles is to develop a software for collaborative clustering [5] with an application to the time series coming from satellite images. One of the objectives of the project is to find new mechanisms to facilitate the labeling (or semantization) of clusters from those images. To achieve this, a proposed solution is to associate textual elements of interest (corresponding to the study themes, and the spatio-temporal perimeter of the time series) with satellite data.

In the project workflow, we permit a user to add constraints to the clustering process in order to improve the results of the latter and to speed-up the process [6]. We help the user to formulate the constraints, for which we use text resources to extract the constraints and we formulate them in the form of rules. For example, the sentence *"Pour être constructible, un terrain, doit avoir un accès à une voie publique ou privée ouverte au public"*[2] contains an obligation with regards to land use, and it can be converted to the following constraint: *"S'il n'y a pas de routes adjacente à la zone constructible* $\rightarrow$ *erreur"*[3].

To allow the automatic extraction of constraints to be implemented, the first identification of potential rules has been done manually by the expert within the documents of interest. These documents come from the thematic expert corpus [7] and have been chosen for their richness in potential rules: they are the written regulations of the Local land plans (PLU – *le Plan Local d'Urbanisme*) and the Natural flood risk prevention plans (PPRI – *le Plan de Prévention des Risques naturels d'Inondation*) of the areas studied.

Not all the textual segments of a selected document can be taken into account. Moreover, some rules have an informative value while others represent a strict constraint. Finally, the application of these rules is not always observable on the satellite data. Therefore, the following classification of textual segments has been defined (Fig. 1).

A text segment is called *pertinent* if it can provide information within the scope of the project: corresponds to our research topics, and contains information in the context of selected research territories. The adequacy of research topics is verified by the presence of words from a nomenclature in the segment. In our application field, remote sensing, the use of nomenclatures and/or ontologies is essential for the labeling process [8], [9]. A *nomenclature* is a collection of thematic words describing the research topics, for example: "chemin de fer" (railroad in English), "stationnement" (parking) and so on. In total, *the Hérelles nomenclature* [4] contains 67 of thematic concepts. All other text that has not been classified as pertinent is considered to be *not pertinent*. These segments may be reminders of the law or definitions, elements that

[2]In English: *"To be authorised for development, land must have an access to a public road or a private road open to the public"*

[3]In English: *"If there are no roads adjacent to the buildable area* $\rightarrow$ *error"*

[4]https://doi.org/10.57745/OXACT8

do not belong to the scope of our study. It includes layout elements, bibliography, headers, footers and so on.

A *strict rule* concerns instructions that have legal force and are therefore enforceable by law. There is no ambiguity in its application (a strict rule clearly states what must be done, what is forbidden, what is allowed).

An *informative rule* refers to segments that provide detailed information about the study topic and the territory. This information is supposed to help understand the results of the proposed solutions. The informative rules also include the segments presented as recommendations.

Some of the strict rules might be difficult to verify, for example with satellite images. We therefore distinguish them by *verifiable* and *non-verifiable*. This distinction is important for the next steps in the project for selecting the constraints.

For more details on the context and concrete examples of the rules please refer to the description of our corpus in [3].

III. Related work

A. Rule identification

Constraint (or rule) extraction or identification can be performed in multiple ways. The majority of approaches in the literature use traditional methods based on bags-of-words and classical Natural Language Processing (NLP) pre-processing. [10] developed a system which automatically detects parts of text describing constraints. A data set is constructed in which words in the sentences are labeled as belonging to constraints or not. To represent words the authors exploit stemmed representation of the words, part-of-speech (POS) tagging and bag-of-words. A machine learning classifier based on Support vector machines (SVM) is employed then to solve the problem. [11] focuses not only on the extraction of constraints, but also grouping them and detecting and displaying relations between constraints. The authors use term frequencies and k-means clustering to achieve their tasks. In the pre-processing step, each document is chunked into sentences and POS tagged. Some constraints are not directly included in sentences. To overcome it, lemmatized representations of words are used. [12] automatically extracts verification constraints from technical documents. The proposed framework is based on 3 core NLP concepts: sentence splitting, tokenization and POS tagging.

In addition to traditional NLP, word embedding [13] can be used to extract information. [14] uses word embedding vectors to derive spatio-temporal characteristics and special indicators from text documents describing food security problems. The method is then used for analyzing the food crisis in West Africa. The problem can also be represented by combining word and image embeddings. [15] demonstrated that both images and text can be mapped into common artificial space then similar vectors can be used to match a caption with an image. [16] solves a similar problem to ours. The authors do not extract constraints explicitly. On the contrary, they map both images and text to a common space. The proposed framework is able to automatically annotate the change images with labels extracted from scientific documents related to the study area. The main disadvantage of this type of modeling is certain lack of control over the process. In this type of approach, it is not possible to intervene in the vector representation and to add other constraints.

The most efficient approach for information extraction is to use an encoder-decoder neural network (NN). This network is pre-trained on a large number of texts to obtain their semantics in the form of high-dimensional vectors. Then, using additional training, the NN can learn a specific task on that representation. This additional training requires a smaller corpus than pretraining because the semantic information has already been acquired during pretraining. The advantage of this approach is the abstraction from the grammar through the use of lexical embeddings. To improve the applicability of the extracted information ontologies can be used on top of that. [17] propose a hybrid approach which uses a NN model to extract constraints and predefined rules to properly extract their relations. This approach is limited to the availability of predefined rules and known relations.

One of the most common encoder-decoder NN is the BERT model [18]. In [19], the authors use encoder-decoder model of type BERT pre-trained on a large number of texts which allows to obtain lexical embeddings to represent semantics as high-dimensional vectors. Another model is then used for learning particular task on a smaller corpus. The problem is represented as a multi-label classification of sentences including constraints or not.

In this work, we experiment with both of the types of approaches: traditional NLP and a state-of-the-art encoder-decoder model. We represent the problem as a text classification task. The classifier which we develop is able to detect constraints in text segments constructed from documents of our interest. Since our documents are in the French language, we are obliged to use one of the BERT extensions for that language. The most common among them are CamemBERT [20] and FlauBERT [21]. The first is a more general model, while the latter better suits for the downstream tasks [22]. In addition, CamemBERT outperforms FlauBERT [22], [23]. We thus will use the former as the state-of-the-art approach implementation.

B. Text augmentation

Data augmentation is usually used for handling lack of data [4], [19] or for improving data imbalance [24]. Text augmentation can be performed in numerous ways from straightforward implementations to using large language models (LLM).

Straightforward approach can include shuffling the words, deleting or replacing random words in the original sentences [25]. These types of methods introduce slight variation in the data, but produce grammatical and syntactical errors. We will test this type of approach in our experiments because of the ease of implementation.

More advanced approach includes replacing selected words by their synonyms derived from specialized dictionaries or by a model of type BERT [18]. New sentences generated using

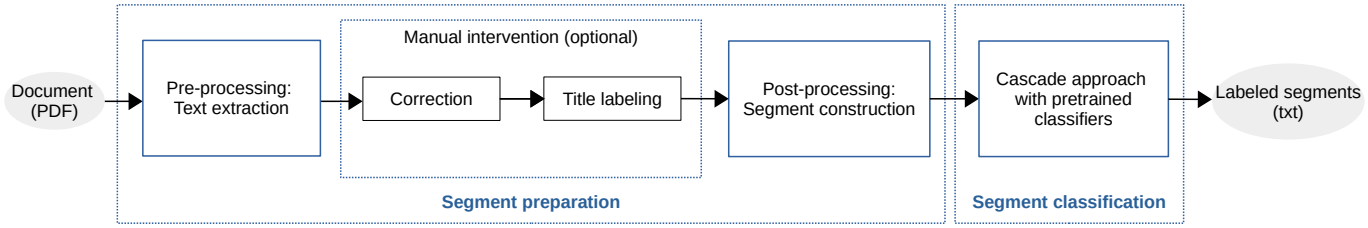

Fig. 2. The workflow presenting different steps in identification and classification of segments of interest in new documents

this method introduce a small variation to the original data and contain most of the linguistic features included in the original sentences. We will implement these types of methods in our work.

The LLM are able to generate semantically similar text without an overlap on the level of words with original phrases. The resulting phrases might be too different with the original data, e.g. not contain important linguistic features. Also, there is not much control over the process and therefore we do not use this type of models in our work.

Last but not least, existing solutions for automatic text augmentation [25] are mainly available for the texts in English only. In case of multi-language models they have numerous limitations, with the length of the input in particular [26]. Few existing works using augmentation of French texts [4] have no publicly available code. We thus have no alternative only to implement our augmentation methods by ourselves[5].

IV. The Proposed Framework

To automatize extraction of the rules from the thematic documents we developed a framework which we refer to as **AIR-FUD (Automatic Identification of Rules in French Urban Documents)**. The AIR-FUD workflow has two main parts: segment preparation and segment classification (Fig. 2). To train a classifier we use a data set which was already constructed in [3]. To construct that data set, 1934 textual segments were manually annotated by the expert as belonging to one of the 4 classes: *Verifiable*, *Non-verifiable*, *Informative* and *Not pertinent*. The details on the data set are presented in Section V below. In the following, we detail how segments are constructed from new documents and which methods are used to perform their classification. In addition, we present text augmentation techniques which we develop for improving the quality of the results.

A. Segment preparation

Segment preparation consists of three steps: text extraction, manual intervention and segment construction (Fig. 2).

1) Text extraction: The both document types in our thematic corpus, the PLU and PPRI, are originally in the Portable Document Format (PDF). We, therefore, extract text from the PDF files of these documents in the pre-processing step first. The output of this step is the set of text fragments in the form of a plain text file. We define a *fragment* as one or several sentences separated by empty lines.

2) Manual intervention: In this step, we manually correct the extracted text. It includes cleaning of the text, for which we remove unnecessary fragments such as the tables of the contents and figure descriptions. In addition, we perform *title labeling*. For that we label all the fragments which are titles and subtitles using sets of special characters as it is described in [3]. Note that this step is optional since our implementation has a fully automatic mode, in which titles and subtitles are extracted automatically from new unseen documents. However, according to our experiments, manual intervention significantly improves the quality of result and thus it is strongly recommended.

3) Segment construction: In post-processing, we perform automatic construction of text segments from labeled fragments. A *segment* in our representation must have a title, subtitle and a rule, while the presence of a sub-subtitle is not mandatory. Sub-subtitles are detected automatically by our segment construction module using a set of predefined patterns. In these patterns, the decision is made by the presence of certain characters in the fragment. A *rule* in our representation is a fragment which is not title, subtitle nor sub-subtitle. The data set that we constructed contains detailed examples of segments constructed from different numbers of fragments [3].

B. Segment classification

Once the segments are constructed, the next step is to perform their classification. To solve it we propose a *cascade* approach which we develop as follows. We split the task into 3 binary classifications applied one by one (Fig. 3). In the *1st classification*, we classify segments by Pertinent (containing the rules) and Not pertinent (all other text). For that we treat the Verifiable, Non-verifiable and Informative classes all together as Pertinent. In the *2nd classification*, we classify segments by Strict (containing strict rules) and Informative (containing not strict rules), for which we treat the Verifiable and Non-verifiable classes as a whole. Finally, we classify the Verifiable and Non-verifiable classes in the *3rd classification*.

For each classification we select a binary classification model which performs best. To make this selection we define and test several baseline methods and a state-of-the-art approach based on Deep Learning. In addition, we develop text augmentation techniques, which we will use to enrich the annotated corpus and improve the results of prediction. In the

[5] The code of our implementations can be found in our framework repository: https://github.com/koptelovmax/AIR-FUD

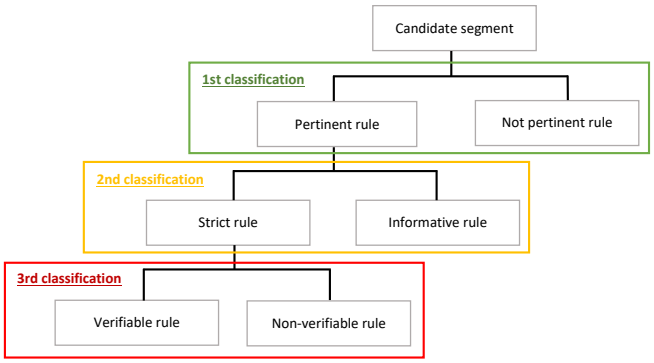

Fig. 3. The cascade classification of hierarchy of classes

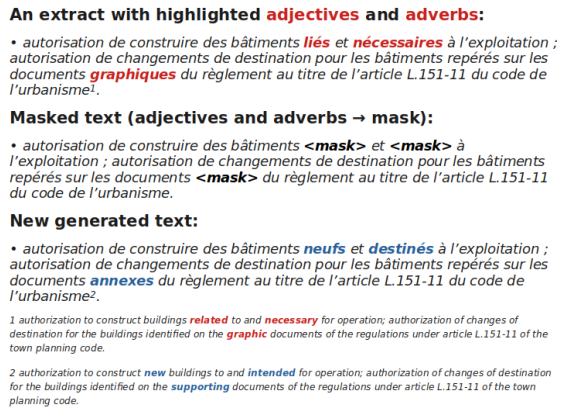

Fig. 4. An example of a text augmentation using CamemBERT and masked word prediction

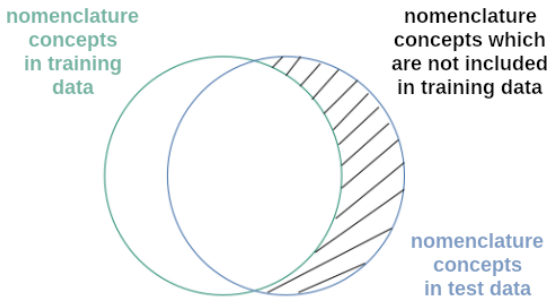

Fig. 5. The Euler diagram representing distribution of nomenclature concepts in training and test data

following we define those methods (Section IV-B1,IV-B2) and these techniques (Section IV-C).

1) Baseline methods:

a) Trigger words: In this method, we exploit a list of *trigger words*, which were extracted from the expert corpus to facilitate the automatic extraction of rules. These words clearly indicate presence of a rule in the fragment or its neighborhood. We have 43 of such words in total, which were built manually by a geographical expert, for example "être interdit" ("is prohibited" in French) or "admettre" ("admit"). The full list can be found in our code[5]. In this method, we first find their stemmed representation. For given examples it will correspond to: "être interd" and "admettr" respectively. Next, we analyze their appearance in segments. By default, all the segments are assigned to the negative class. We check whether a trigger is present in a neighborhood of segments of size n. If yes, all of these segments are considered to be the positive class.

b) Vector similarity model: For the pre-processing, we perform tokenization of segments, remove stop words and get a stemmed representation of the rest. We use the result to compute the Term Frequency (TF) [27]:

$$TF(t,d) = \frac{freq(t,d)}{\sum_m freq(t,d)},$$

where $freq(t,d)$ – frequency of term t in segment d, m – total number of terms, and the TF-Inverse Document Frequency (TF-IDF) [28], [29]:

$$TF\text{-}IDF(t,d) = TF(t,d) \cdot \log \frac{N}{df(t)},$$

where N – total number of segments, $df(t)$ – number of segments containing t. We use both frequencies to construct the frequency vectors. To achieve that we represent each segment d by a vector of term frequencies, $F(d)$, having size m. Each element t of the segment in this vector corresponds to $TF(t,d)$ or $TF\text{-}IDF(t,d)$. We then use the resulting vectors for solving the binary classification task, which is modeled as follows. When new segment d_{new} arrives, we compute mean of its similarity with all segments of the positive class, then with all segments of the negative class:

$$sim_{class}(d_{new}) = mean\Big(\sum_{i \in \{d_{class}\}} F(d_{new}) \times F(i)\Big),$$

where d_{class} – segments labeled as $class$. If $sim_{pos}(d_{new}) > sim_{pos}(d_{new})$, d_{new} receives the positive class and negative otherwise.

c) Machine learning (ML) using frequency vectors: In this method, we use the same vectors $F(d)$ to represent segments as in the previous method. The difference is that this time we employ a machine learning model to learn a binary classifier. When a new segment d_{new} arrives, we classify it with the trained model.

2) State-of-the-art approach: Following our discussion in section III we employ CamemBERT as the state-of-the-art approach for text classification. In this approach, we fine-tune the original CamemBERT model for the binary classification task using labeled segments from our data set. Each unseen segment we classify then with the fine-tuned model.

C. Text augmentation

In order to improve the results of the CamemBERT model we perform augmentation of the training data, which then are used for fine-tuning the model. We use each segment k times to generate k new segments. Using this method, we increment

TABLE I
NUMBER OF CONCEPTS IN THE NOMENCLATURE, EXPERT NOMENCLATURE AND THEIR EXTENDED VERSIONS

Type of nomenclature	Number of concepts
Nomenclature	67
Enriched nomenclature (WordNet, $s = 5$)	134
Enriched nomenclature (Agrovoc, $s = 5$)	120
Enriched nomenclature (DES, $s = 5$)	153
Expert nomenclature	207
Enriched expert nomenclature (WordNet, $s = 5$)	406
Enriched expert nomenclature (Agrovoc, $s = 5$)	429
Enriched expert nomenclature (DES, $s = 5$)	487

the number of examples of an underrepresented class, which improves the imbalance of our data. In the following, we detail different strategies which we use to generate new text. They are based on grammatical information, semantic information, etc.:

- **POS-driven method**: In this method, we replace certain words in each segment by semantically meaningful phrases derived by CamemBERT. To achieve that we mask certain parts of speech in the segment, then ask CamemBERT to solve the masked word prediction task, a main principle of language models of type BERT [18]. In this task, the model tries to predict the original vocabulary of the masked content based only on its context. As for the selection of parts of speech, we mask all adjectives and adverbs as in [4], because they are usually not part of thematic key phrases. The aim of this method is to introduce a variety in newly generated segments without changing the main content. In practice, newly generated sentences are grammatically correct, but not always have the same meaning (Fig. 4). We also experiment with masking all verbs and all nouns in the segments to generate more diverse examples.
- **Semantic-driven method**: This method is based on the hypothesis that an ideal classifier would classify segments by presence of words from the nomenclature. In this hypothesis, we assume that not all nomenclature words which are included in the test data are present in the training data, and thus we can artificially include them into the training data (Fig. 5). To achieve that, we replace a random word in the segment by a random concept from an *enriched nomenclature*. Since we do not have the full list of nomenclature concepts, but only 67 thematic words, we use a special dictionary to enrich it with synonyms. We select s synonyms at most for such a dictionary (Table I). A new segment generated by this method is not always grammatically correct, but it guaranteedly includes at least one nomenclature concept from an enriched vocabulary (Fig. 6).
- **Combined approach**: This method is based on the two previous ideas. First, we check the presence of the words from an extended *expert nomenclature*[6] in the segment (Table I). If at least one of these words is present in the segment, we use the latter to generate a new segment with the POS-driven method.

[6]A nomenclature, manually extended by an expert (included in our code)

Fig. 6. An example of a text augmentation by replacing a random word by a random nomenclature concept

V. EXPERIMENTAL EVALUATION

A. Data set

Our data set [3] which we use for experiments contains 1934 labeled segments extracted from 9 the PLU and PPRI documents. In the data, the segments are labeled as belonging to one of 4 classes: Verifiable, Non-verifiable, Informative and Not pertinent. We combine the class Verifiable and Non-verifiable to derive the class *Strict*, and we combine the class Strict and Informative to derive the class *Pertinent* (Fig. 1). The detailed statistics on each type of segment and each document are presented in Table II.

B. Experimental settings

1) Model parameters: We perform evaluation of our methods using the following parameters. In trigger words, we fix $n \in [1..10]$. In the vector similarity model, we use two types of term frequencies: TF and TF-IDF. In the ML method, we use two types of vectors: based on TF and TF-IDF. In addition, we experiment with 4 classifiers: Decision Trees [30], Random Forests [31], SVM [32] and Stochastic Gradient Descent (SGD) [33]. We only report best parameter setting w.r.t. each baseline method.

In the state-of-the-art implementation, we use the parameters recommended in [18]: learning rate $2 \cdot 10^{-5}$ and $\epsilon = 10 \cdot 10^{-8}$. We also fix the number of epochs to 10 and the batch size to 16. We repeat each experiment 10 times to address the model instability problem [34] and report best and average results, which we compute using the mean function.

As for data augmentation, we test each of the methods presented in Section IV-C with $k \in [1..5]$. As before, we only report results corresponding to best performing values of k. To implement the Semantic-driven method and Combined approach we test 3 different dictionaries as a source of synonyms: WordNet [35], Agrovoc [36] and DES [37]. For each of the dictionaries we fix $s = 5$ based on our preliminary experiments (Table I).

TABLE II
NUMBER OF SEGMENTS CORRESPONDING TO EACH CLASS AND EACH DOCUMENT

Document	Number of segments								
	1st classification			2nd classification			3rd classification		
	Pertinent	Not pertinent	Total	Strict	Informative	Total	Verifiable	Non-verifiable	Total
PLU Montpellier ZONE-A	29	42	71	27	2	29	8	19	27
PLU Montpellier ZONE-N	48	78	126	39	9	48	12	27	39
PLU Montpellier ZONE-AU0	31	58	89	28	3	31	6	22	28
PLU Montpellier ZONE-14AU	23	59	82	21	2	23	8	13	21
PLU Montpellier ZONE-5AU	30	64	94	27	3	30	4	23	27
PLU Montpellier ZONE-4AU1	65	101	166	55	10	65	7	48	55
PPRI Montpellier	88	37	125	83	5	88	22	61	83
PPRI Grabels	54	45	99	47	7	54	33	14	47
PLU Grabels	306	776	1082	261	45	306	47	214	261
Total	674	1260	1934	588	86	674	147	441	588
	35%	65%	100%	87%	13%	100%	25%	75%	100%

2) Evaluation protocol: We do not require any specific validation framework for trigger words since there is no training phase in the method. To evaluate the vector-similarity model we use *leave-one-out* cross validation (CV) implemented as follows: each of the segments is used for testing while all the others are used for training. For the ML method we use a more common validation framework. To perform evaluation of this method we implement *10-fold* CV, with each fold containing 10% of all segments. The model is trained on 9 folds while the last fold is used for validation. The process is repeated 10 times until each fold is used as a test set.

Finally, to evaluate the state-of-the-art approaches we use *stratified* CV implemented as follows. The data are split into 2 parts: 80% of segments are used for learning, while the other 20% are used for validation. The split is performed in such a way that the proportion of positive and negative examples for each type of classification remains the same (Table II)

3) Quality measures: In the 1st classification, we use Precision, Recall and F_1 score to assess the quality of our prediction:

$$Prec = \frac{TP}{TP + FP},$$

$$Rec = \frac{TP}{TP + FN},$$

$$F_1 = 2 \cdot \frac{Prec \cdot Rec}{Prec + Rec},$$

where TP – true positive examples, FP – false positive and FN – false negative. In the 2nd and 3rd classification, we compute each of those values for both of the classes. To determine which of the results is the best we use *weighted* accuracy. For that we assign a classification cost of 1 to examples of an over-represented class (Strict and Non-verifiable) and cost new_cost to examples of an underrepresented class (Informative and Verifiable), derived by:

$$new\_cost = \frac{|D|}{2 \cdot |N|},$$

where $|D|$ – number of examples of both classes for the classification task, $|N|$ – number of examples of an underrepresented class. We then perform evaluation based on the costs defined: FN and TN receive score new_cost for every example of an underrepresented class w.r.t. its real class, while FP and TP receive score 1 for positives. We benefited from using weighted accuracy twice: to determine the best performing epoch in each experiment and to select the best result among 10 runs.

TABLE III
EVALUATION RESULTS WITH DIFFERENT METHODS ON 1ST CLASSIFICATION TASK

Method	Results		
	Precision	Recall	F_1 score
Trigger words (n = 10)	0.35	0.98	0.51
Vector similarity (TF-IDF)	0.66	0.96	0.78
ML approach (TF, SVM)	0.87	0.81	0.83
CamemBERT	0.86	0.96	0.91

4) Implementation details: We implemented baseline methods, the state-of-the-art and text augmentation approaches in Python[5]. We used the NLTK libary [38] to implement tokenization, remove stop words and find stemmed representation of segments for the baseline methods. In addition, we used the Stanford POS-Tagger [39] for the French language to determine part-of-speech in the POS-driven method and Combined approach for data augmentation. We used the scikit-learn library [40] to implement Decision Trees, Random Forests, SVM and SGD classifiers. We also used this library to implement precision, recall, F_1 score and weighted version of accuracy. Finally, we used the *CamembertForSequenceClassification* model from the HuggingFace library [41] as the CamemBERT implementation.

C. Results

1) 1st classification: The results are presented in Table III. Trigger words are able to discover almost all pertinent segments (recall 98%). However, the low precision of this method results in the average performance equal to random guessing (F_1 score 51%). The vector similarity method demonstrates identical recall, but improves on precision twice compared to trigger words. The ML approach improves further on precision, but its recall drops compared to the previous method. Nevertheless, it slightly outperforms the latter. Finally, CamemBERT smooths out this difference by providing

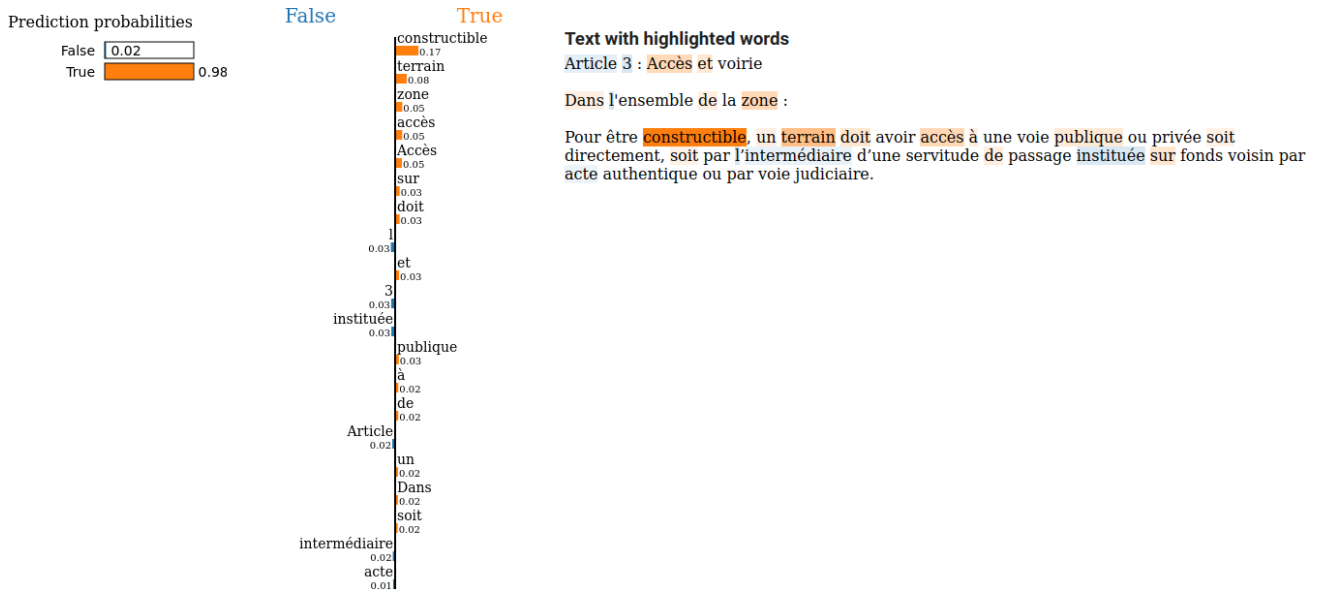

Fig. 7. An example of a segment analysis using Lime, from left to right: the classification results with probability scores, the features which led to these results, original segment with negative and positive descriptors highlighted in blue and orange consequently (with some of the descriptors coming from the expert nomenclature concepts: *terrain*, *zone* and *accès*)

TABLE IV
EVALUATION RESULTS WITH DIFFERENT METHODS ON 2ND CLASSIFICATION TASK

Method	Results						
	Class Strict			Class Informative			Accuracy*
	Precision	Recall	F_1 score	Precision	Recall	F_1 score	
Trigger words (n = 1)	0.92	0.56	0.70	0.18	0.67	0.29	0.60
Vector similarity (TF-IDF)	0.99	0.92	0.95	0.62	0.92	0.74	0.92
ML approach (TF-IDF, SGD)	0.97	0.99	0.98	0.93	0.81	0.85	0.93

precision similar to the ML approach and recall identical to the vector similarity method. The overall result of CamemBERT (F_1 score 91%) demonstrates very good performance of the method. We will thus use the classifier trained by this method in our framework.

In order to verify the quality of the resulting classifier, we perform a detailed study of the segments classified by CamemBERT. For each example in the test data classified as TP[7] we collect all features which led to this result using *Lime* [42] (Fig. 7). As a result, we find out that 35.06% of all positive features are the expert nomenclature concepts. This is a very good result showing that CamemBERT is able to capture thematic concepts and use them as indicators of positive examples. Another result of this analysis is that 10.0% of all top 1 distinct features are trigger words. Despite the relatively low percentage, this is also a good result given that the features should not only consist of nomenclature concepts.

2) 2nd classification: The results are shown in Table IV. Trigger words provide satisfactory performance for the class Strict (F_1 score 70%) and quite a low result for the Informative class. It can be explained by the fact that trigger words are included in both of the classes, which makes this method ineffective. The vector similarity method improves on the results for both of the classes, however the class Strict outperforms Informative due to high imbalance of classes (Table II). The ML approach compensates this shortcoming by improving the results of the underrepresented class. The overall performance of the resulting classifier is good enough (weighted accuracy 93%) which makes us choose it for the framework. Trying to further improve the results might cause overfitting and we thus do not employ the state-of-the-art for this task.

3) 3rd classification: The results are presented in Table V. Trigger words perform similar to the 2nd classification task with the only difference that the class Verifiable has worse results than the class Non-verifiable. The two next methods, Vector similarity and the ML approach, improve the results of Trigger words keeping the trend of better performance of the Non-verifiable class. The latter can be explained by the fact that the Non-verifiable class is better represented in the data than the Verifiable class (Table II). CamemBERT slightly improves the situation by minimizing this difference to 13% (F_1 class Verifiable 82% vs F_1 class Non-verifiable 94%). We try to minimize this difference further by performing data augmentation of the underrepresented class (Section III-B). To achieve that we continue with applying our methods defined in Section IV-C.

[7]128 segments are classified as TP out of 135 positive examples in the test data

TABLE V
EVALUATION RESULTS WITH DIFFERENT METHODS ON 3RD CLASSIFICATION TASK

Method	Results						
	Class Verifiable			Class Non-verifiable			Accuracy*
	Precision	Recall	F_1 score	Precision	Recall	F_1 score	
Trigger words (n = 1)	0.31	0.70	0.43	0.83	0.49	0.62	0.57
Vector similarity (TF-IDF)	0.53	0.97	0.68	0.98	0.71	0.83	0.81
ML approach (TF, Decision Trees)	0.66	0.78	0.71	0.94	0.89	0.91	0.85
CamemBERT	0.78	0.86	0.82	0.95	0.92	0.94	0.90
CamemBERT+data augmentation	0.82	0.93	0.87	0.98	0.93	0.95	0.93

TABLE VI
RESULTS OF TEXT AUGMENTATION FOR 3RD CLASSIFICATION AND THEIR COMPARISON WITH PERFORMANCE ON ORIGINAL DATA

Method	Size of training data			Results on test data				
				Class Verifiable		Class Non-verifiable		Accuracy*
	Number of segments		% positive	F_1 score		F_1 score		
	Total	Positive		avg	max	avg	max	
Original data	470	118	25%	0.80	0.82	0.93	0.94	0.90
POS-driven method (adj+adv, k=1)	573	221	39%	0.82	0.86	0.94	0.96	0.92
POS-driven method (nouns, k=2)	706	354	50%	0.81	0.84	0.94	0.95	0.91
POS-driven method (verbs, k=3)	818	466	57%	0.83	0.86	0.94	0.95	0.92
Semantic-driven method (WordNet, k=1)	588	236	40%	0.82	0.85	0.93	0.95	0.92
Semantic-driven method (DES, k=2)	706	354	50%	0.82	0.87	0.94	0.95	0.93
Semantic-driven method (Agrovoc, k=3)	824	472	57%	0.82	0.86	0.94	0.95	0.92
Combined approach (DES, adj+adv, k=1)	540	188	35%	0.82	0.87	0.94	0.95	0.93
Combined approach (DES, nouns, k=2)	628	276	44%	0.82	0.87	0.94	0.96	0.93
Combined approach (DES, verbs, k=4)	786	434	55%	0.83	0.85	0.94	0.95	0.92

4) Text augmentation: Most of the best results on data augmentation correspond to the settings with the positive class[8] augmented to 50% and more (Table VI). This correlation can be noticed mostly in the POS-driven method and the Semantic-driven method (class Verifiable max F_1 score 86-87%). In spite of that, the Combined approach allows to keep the data unbalanced and have the same (or even better[9]) performance. Surprisingly enough, the Semantic-driven method and Combined approach are able to improve the results. According to our initial hypothesis, all pertinent segments should include nomenclature concepts. Following the results on augmented data, we can make a conclusion that the class Verifiable contains more nomenclature concepts than the class Non-verifiable. Finally, the best result in our experiments corresponds to the Combined approach with the DES dictionary as a source of synonyms (Table V). The setting with replacing adjectives and adverbs requires fewer cycles of data augmentation ($k = 1$) compared to the setting with nouns, which requires repeating the augmentation process twice ($k = 2$). Using this method we are able to improve the performance of CamemBERT for the Verifiable class by 6% (max F_1 score from 82% to 87%). We will thus use the resulting method in the AIR-FUD framework for processing new documents.

VI. CONCLUSION AND FUTURE DIRECTIONS

In this work, we presented the AIR-FUD framework for (semi-)automatic identification of rules in urban planning documents in the French language. This framework aims to address the needs of the Hérelles project. We showed experimentally using manually annotated corpus that our framework is able to correctly classify the rules with the hierarchy of classes. We proposed a cascade approach for that and we demonstrated a good performance of the latter. In addition, we developed several text augmentation methods based on text mining and a language model which are able to solve the data imbalance problem and improve the overall results for the latest classification task.

As for the future work, we aim to apply our framework to new unseen documents. Based on the results, we might fix the choice of classifiers and the data augmentation method for each of the classification tasks. Next, we intend to continue the work on the project by extracting the constraints (e.g. in the form of "if... then...") from the segments classified as pertinent. We plan to investigate named entity recognition [43] and abstractive summary generation [44] for that.

[8]the class Verifiable in our case

[9]taking into account weighted accuracy

ACKNOWLEDGMENT

This work is partially funded by the ANR project Hérelles ANR-20 CE23-0022.

REFERENCES

[1] "Artificialised land and artificialisation processes: determinants, impacts and levers for action," https://www.inrae.fr/en/news/artificialised-land-and-artificialisation-processes, published: 2017-12-08. Accessed: 2023-03-13.

[2] J. Daniell, F. Wenzel, and A. Schaefer, "The economic costs of natural disasters globally from 1900-2015: historical and normalised floods, storms, earthquakes, volcanoes, bushfires, drought and other disasters," in *EGU general assembly conference abstracts*, 2016, pp. EPSC2016–1899.

[3] M. Holveck, M. Koptelov, M. Roche, and M. Teisseire, "Segments textuels Hérelles," 2023. [Online]. Available: https://doi.org/10.57745/DWYGMB

[4] A. Laifa, L. Gautier, and C. Cruz, "Impact of textual data augmentation on linguistic pattern extraction to improve the idiomaticity of extractive summaries," in *Big Data Analytics and Knowledge Discovery: 23rd International Conference, DaWaK 2021, Virtual Event, September 27–30, 2021, Proceedings 23*. Springer, 2021, pp. 143–151.

[5] A. Cornuéjols, C. Wemmert, P. Gançarski, and Y. Bennani, "Collaborative clustering: Why, when, what and how," *Information Fusion*, vol. 39, pp. 81–95, 2018.

[6] I. Davidson and S. Ravi, "Agglomerative hierarchical clustering with constraints: Theoretical and empirical results," in *Knowledge Discovery in Databases: PKDD 2005: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005. Proceedings 9*. Springer, 2005, pp. 59–70.

[7] R. Kafando, R. Decoupes, M. Teisseire, L. Sautot, and C. Weber, "Constitution de corpus thématique: Pour un meilleur suivi du territoire de la métropole de montpellier méditerranée," in *SAGEO'21 16ème Conférence Internationale de la Géomatique, de l'Analyse Spatiale et des Sciences de l'Information Géographique.*, 2021.

[8] H. Luo, L. Li, H. Zhu, X. Kuai, Z. Zhang, and Y. Liu, "Land cover extraction from high resolution zy-3 satellite imagery using ontology-based method," *ISPRS International Journal of Geo-Information*, vol. 5, no. 3, p. 31, 2016.

[9] M. Alirezaie, A. Kiselev, M. Längkvist, F. Klügl, and A. Loutfi, "An ontology-based reasoning framework for querying satellite images for disaster monitoring," *Sensors*, vol. 17, no. 11, p. 2545, 2017.

[10] Z. Kiziltan, M. Lippi, P. Torroni *et al.*, "Constraint detection in natural language problem descriptions," in *IJCAI*, vol. 2016. International Joint Conferences on Artificial Intelligence, 2016, pp. 744–750.

[11] K. Winter and S. Rinderle-Ma, "Detecting constraints and their relations from regulatory documents using nlp techniques," in *On the Move to Meaningful Internet Systems. OTM 2018 Conferences: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part I*. Springer, 2018, pp. 261–278.

[12] M. W. Anwar, I. Ahsan, F. Azam, W. H. Butt, and M. Rashid, "A natural language processing (nlp) framework for embedded systems to automatically extract verification aspects from textual design requirements," in *Proceedings of the 2020 12th International Conference on Computer and Automation Engineering*, 2020, pp. 7–12.

[13] F. Almeida and G. Xexéo, "Word embeddings: A survey," *arXiv preprint arXiv:1901.09069*, 2019.

[14] C. T. Ba, C. Choquet, R. Interdonato, and M. Roche, "Explaining food security warning signals with youtube transcriptions and local news articles," in *Proceedings of the 2022 ACM Conference on Information Technology for Social Good*, 2022, pp. 315–322.

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[16] N. Neptune and J. Mothe, "Automatic annotation of change detection images," *Sensors*, vol. 21, no. 4, p. 1110, 2021.

[17] C. Wu, P. Wu, J. Wang, R. Jiang, M. Chen, and X. Wang, "Developing a hybrid approach to extract constraints related information for constraint management," *Automation in Construction*, vol. 124, p. 103563, 2021.

[18] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[19] A. Remaud, "Extraction de contraintes dans des spécifications de validation de données," *Extraction et Gestion des Connaissances: EGC'2022*, vol. 38, 2022.

[20] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de La Clergerie, D. Seddah, and B. Sagot, "Camembert: a tasty french language model," *arXiv preprint arXiv:1911.03894*, 2019.

[21] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "Flaubert: Unsupervised language model pre-training for french," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 2479–2490.

[22] Y. Guo, V. Rennard, C. Xypolopoulos, and M. Vazirgiannis, "Bertweetfr: Domain adaptation of pre-trained language models for french tweets," in *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, 2021, pp. 445–450.

[23] E. Kelodjoue, J. Goulian, and D. Schwab, "Performance of two french bert models for french language on verbatim transcripts and online posts," in *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, 2022, pp. 88–94.

[24] S. Afzal, M. Maqsood, F. Nazir, U. Khan, F. Aadil, K. M. Awan, I. Mehmood, and O.-Y. Song, "A data augmentation-based framework to handle class imbalance problem for alzheimer's stage detection," *IEEE access*, vol. 7, pp. 115 528–115 539, 2019.

[25] P. Damodaran, "Parrot: Paraphrase generation for nlu," 2021. [Online]. Available: https://github.com/PrithivirajDamodaran/Parrot_Paraphraser

[26] "Rasa paraphraser," 2021. [Online]. Available: https://github.com/RasaHQ/paraphraser

[27] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[28] A. Bougouin, S. Barreaux, L. Romary, F. Boudin, and B. Daille, "Termith-eval: a french standard-based resource for keyphrase extraction evaluation," in *LREC-Language Resources and Evaluation Conference*, 2016.

[29] M. Liang and T. Niu, "Research on text classification techniques based on improved tf-idf algorithm and lstm inputs," *Procedia Computer Science*, vol. 208, pp. 460–470, 2022.

[30] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81–106, 1986.

[31] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[33] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[34] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi, "Revisiting few-sample bert fine-tuning," in *Proceedings of the Ninth International Conference on Learning Representations (ICLR)*, 2020.

[35] G. A. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.

[36] C. Caracciolo, A. Stellato, A. Morshed, G. Johannsen, S. Rajbhandari, Y. Jaques, and J. Keizer, "The agrovoc linked dataset," *Semantic Web*, vol. 4, no. 3, pp. 341–348, 2013.

[37] M. Morel and J. François, "Le dictionnaire electronique des synonymes du crisco: un outil de plus en plus interactif," *Revue française de linguistique appliquée*, vol. 20, no. 1, pp. 9–28, 2015.

[38] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[39] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 2003, pp. 252–259.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[41] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2020, pp. 38–45.

[42] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[43] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[44] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, 2013, pp. 178–186.