Université Jean Monnet

and

Comparison of named entity recognition methods for geographical information retrieval

June 26, 2024

Internship Report

Hedi Zeghidi

Supervised by:
Ludovic Moncla, INSA Lyon, LIRIS UMR CNRS 5205
Marc Sebban, Université Jean Monnet, Saint-Etienne

Contents

1 Introduction

Today, knowledge with all the new canal of information such as the television and the internet have become available to all and each subject you have an interest in is explained somewhere. But in the past, there was already some encyclopedia that sought to present readers with a synthesis of the knowledge of his time. This internship will focus on identifying the semantic representation of entities evoked such as spatial entities (cities, rivers, mountains, etc.), person entities (name, title, etc.), the relation entity, and the geographic coordinates (latitude/longitude) in different encyclopedia such the Encyclopédie by Diderot, d'Alembert, and Jaucourt. We study multiple methods to identify the entities and we train and compare them using a gold standard dataset[1] [9].

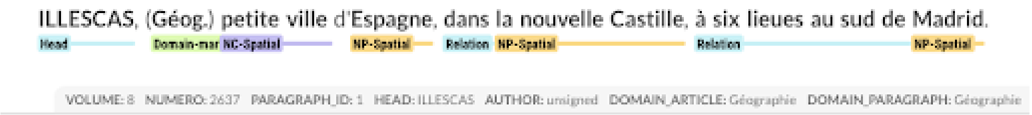
As part of the GEODE project[2], a first model was trained with an active learning method, using the Prodigy web-based tool which resulted in a custom spaCy spancat model. It demonstrates strong overall performance, achieving an F-score of 86.42%. Evaluations for each span category reveal strengths in recognizing spatial entities and persons (including nominal entities, named entities, and nested entities). This will be used as a baseline. However, the model has difficulty with certain classes of the dataset such as the longitude and latitude, the miscellaneous (MISC) entities, or the MISC nested entity. So, the objective is to experiment and evaluate other architectures such as Transformers, a BI-LSTM model, a CNN model and a Generative Pre-trained Transformer.

We have considered different levels of difficulty for our different models by varying the entities to identify:

1. Named entities are specific elements within the text that represent unique entities, such as people's names, locations, organizations, dates, and so on which are all on the same level and there are no intersections or overlapping between them (as presented in the Figure 1). It focuses primarily on identifying named entities within the text rather than analyzing their relationships or roles within a larger context. It provides a foundational level of information extraction that can be useful in various Natural Language Processing (NLP) applications such as information retrieval, document categorization, and sentiment analysis.

2. Nested entities of level 1 refers to the identification and classification of named entities within a text while also recognizing the entities which can contain other named entities as components. In the this example, we focus on the nested entities with only one level of imbrication. For example in the Figure 2, we have an entity "ENE-Spatial" which is composed of two entities "NC-Spatial" and "NP-Spatial".

3. Nested entities of all levels refers to the same parameter all before but with all the nested entities of all levels such as in the Figure 3 where we have three levels of entities "ENE-Spatial".



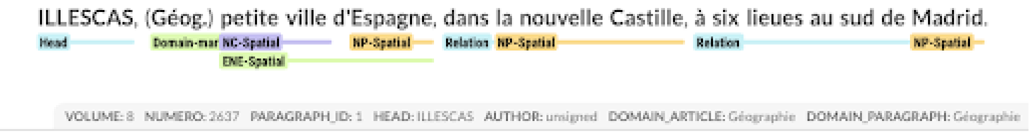
ILLESCAS, (Géog.) petite ville d'Espagne, dans la nouvelle Castille, à six lieues au sud de Madrid.

VOLUME: 8 NUMERO: 2637 PARAGRAPH_ID: 1 HEAD: ILLESCAS AUTHOR: unsigned DOMAIN_ARTICLE: Géographie DOMAIN_PARAGRAPH: Géographie

Figure 1: Example of NER annotations (with no overlaps)

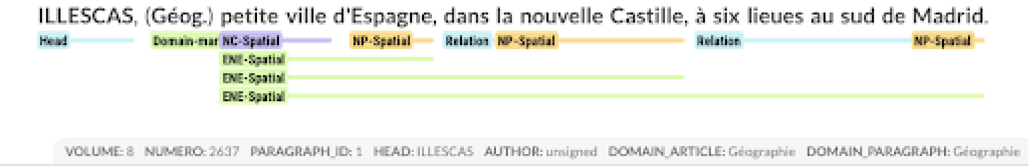
[1] https://huggingface.co/datasets/GEODE/GeoEDdA

[2] https://geode-project.github.io


ILLESCAS, (Géog.) petite ville d'Espagne, dans la nouvelle Castille, à six lieues au sud de Madrid.

VOLUME: 8 NUMERO: 2637 PARAGRAPH_ID: 1 HEAD: ILLESCAS AUTHOR: unsigned DOMAIN_ARTICLE: Géographie DOMAIN_PARAGRAPH: Géographie

Figure 2: Example of spans recognition (with nested entities up to level 1)


ILLESCAS, (Géog.) petite ville d'Espagne, dans la nouvelle Castille, à six lieues au sud de Madrid.

VOLUME: 8 NUMERO: 2637 PARAGRAPH_ID: 1 HEAD: ILLESCAS AUTHOR: unsigned DOMAIN_ARTICLE: Géographie DOMAIN_PARAGRAPH: Géographie

Figure 3: Example of span recognition (with nested entities)

2 Research Lab, Team and Project

The research lab that hosted me was the Laboratoire d'InfoRmatique en Images et Systèmes d'Information (LIRIS[3]). LIRIS has four units in Lyon, one at the INSA Lyon in Villeurbanne, a second one at the University Claude Bernard Lyon 1, a third one at Ecole centrale de Lyon, and one at University Lumière Lyon 2. This laboratory was created in 2003 and focuses on Computer Science and more generally Information Science and Technology, and today it has 330 members divided into 12 teams.

This lab focuses on 6 areas of expertise which are:

1. Data, Systems, and Security (BD, DRIM, SOC, and DM2L teams)
2. Computer Graphics, and Geometry (ORIGAMI team)
3. Images, Vision, and Learning (IMAGINE team)
4. Interaction and Cognition (SICAL, SyCoSMA, and TWEAK teams)
5. Algorithms and Combinatorics (GOAL team)
6. Simulation and Life Sciences (teams SAARA and BEAGLE)

During these 6 months, I work in the DM2L team with Ludovic Moncla (Associate professor at INSA Lyon) on the project GEODE with at least one meeting per week. The project GEODE funded by LabEX ASLAN has the objective of studying changes in geographical discourse between 1750 and the present in a corpus of four French encyclopedias. To this end, we use methods of semi-supervised text classification, language model generation, and automatic discourse routine detection. For this project, we used Python with multiple packages such as PyTorch, Flair, Transformers, scikit-learn, CRF, Spacy, and Matplotlib. While working on this project, I also learn how to use a Pagoda server[4] to run my program with the help of Olivier MBAREK (Responsable Technique of the PAGODA Plateform at LIRIS). Furthermore, I attended DM2L team seminars including preparations for thesis presentations and GEODE seminars such as "Traitement de données complexes en Géographie" done by Helen Rawsthorne where she presented how to create a geospatial knowledge graph from text[5].

[3] https://liris.cnrs.fr/
[4] https://projet.liris.cnrs.fr/pagoda/latest/
[5] https://gitlab.liris.cnrs.fr/geode/seminaires-ixxi/-/tree/master/sÃľminaires/session19_mar24

3 Related Works

NLP lies at the intersection of computer science and information retrieval, focusing on empowering computers to understand and interact with human language [7]. This field encompasses the analysis of textual and spoken data sets, employing rule-based or probabilistic (including statistical and cutting-edge neural network) machine learning techniques. One of the tasks of NLP is Named Entity Recognition (NER) where we study the extraction of various types of entities and the structuring of information from unstructured data. And so, we fine-tune different family of models on the dataset such as a Rule-Based models, Large Language Models (LLMs) or Machine Learning models.

3.1 Rule-Based Models

A rule-based model (such as in [1]) identifies and classifies entities such as names of people, organizations, locations, dates, and other proper nouns using a predefined set of rules. These rules might include specific patterns like capitalization (e.g., capitalized words at the start of a sentence often being names), known lists of entity names (dictionaries), and contextual cues (e.g., "Dr." followed by a capitalized word likely indicating a person's name). For example, a rule might state that any capitalized word following "Mr." or "Ms." is likely a person's name. Rule-based NER systems are highly interpretable, as each entity recognition decision can be traced back to specific rules.

3.1.1 Perdido

Perdido is a Python library designed for geoparsing French texts [8]. Geoparsing, a crucial task in geographic information retrieval and NLP, encompasses two primary subtasks: (1) recognition and classification of named entities and spatial information (also known as geotagging), and (2) toponym resolution (also referred to as geocoding). The Perdido Geoparser is structured into three layers: a back-office component hosted on a server, a REST API that exposes the functionalities of the back-office as web services, and a Python library that provides an additional layer for querying the services, manipulating, visualizing, and exporting the results. We will used the output produced by the NER step as a baseline.

3.2 Machine and Deep Learning

When talking about Machine and Deep Learning, we have different types of architectures, each having their special mechanisms, and different approaches such as the classical Conditional Random Field (CRF) with the use of the Markov property or the more recent Transformers with the attention-mechanisms.

3.2.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of deep learning model for processing grid-like data, such as images or text. CNNs use convolutional layers to automatically and adaptively learn spatial hierarchies of features from input data. These networks are adept at identifying patterns and local dependencies in text, making them well-suited for the task of recognizing entities like names, dates, and locations [12]. The model follows a transition-based approach, which involves making a sequence of local decisions to label the text accurately. For instance, the well known and widely used spaCy NER Python package[6] relies on CNNs which have pre-trained models with a predefined set of entities such as "fr_core_news_sm" which detected LOC, MISC, ORG, and PER for French. The spaCy framework also provides tools for training our own model with our own defined set of entities.

[6] https://spacy.io

3.2.2 Conditional Random Field

CRF are a type of Markov Random Field (MRF), which is a set of random variables having a Markov property described by an undirected graph such as in the Figure 4 where all the edges between two variables represent their dependency, for example A depend on B and D. CRF are a class of probabilistic graphical model, which use the observed data to predict the labels of a sequence, while taking into account the dependencies between neighboring labels [5]. Each token has a set of parameters, but also the set of the parameters of the previous token and the next token. In our case, the base set of parameters of each token are:

1. Token: Word form
2. lower: Lowercase word form
3. isdigit: True if word is a number else False
4. isupper: True if word is all capital letter else False
5. ispunct: True if word is punctuation else False
6. isstop: True if word is empty else False
7. len: Number of characters composing the word
8. shape: Shape of the word where capital letter becomes *'X'*, lowercase letter *'x'* , any digits is replaced by *'d'*, and punctuation is *'.'* (for example, France become *'Xxxxxx'*)
9. pos: Grammar part of the sentence (NOUN, VERB, etc)
10. dep: Syntactic role of the word

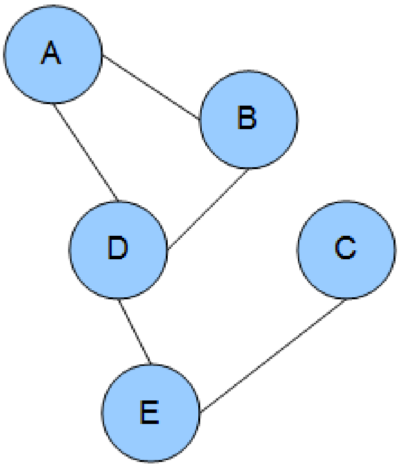


Figure 4: Markov Random Field (source Wikipedia)

3.2.3 Bi-LSTM

Bi-LSTM stands for Bidirectional Long Short-Term Memory. It is a type of recurrent neural network that is particularly effective for sequence-based tasks. A Bi-LSTM combines two LSTMs: one processing the input sequence from start to end (forward direction) and another processing the sequence from end to start (backward direction). In [6], the authors utilized the Flair package to train and evaluate a Bi-LSTM model. This model consists of three layers. The first layer is a contextual embedding with a vector dimension of 300, created using Flair's StackedEmbeddings function, which combines a FastText embedding (contextualized by the following token) and another embedding contextualized by the preceding token. These embeddings were trained using data from the French Wikipedia. The model's second and third layers are LSTM layers, each with a word dropout rate of 0.05, followed by a Linear layer whose dimension matches the number of entities.

3.3 Large Language Models (LLMs)

Large Language Models (LLMs) are advanced artificial intelligence systems designed to understand and generate human language. These models, such as OpenAI's GPT, are trained on vast amounts of text data, enabling them to predict and generate text based on given inputs. They leverage deep learning techniques, particularly transformers, to capture the context and nuances of language, allowing them to perform a wide range of tasks, including text completion, translation, summarization, and questions answering.

3.3.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) [2] is a pre-trained transformer-based model introduced by Google in 2018. It revolutionized natural language understanding tasks by leveraging bidirectional context to capture deeper semantic understanding. BERT is trained on large amounts of text data using masked language modeling, subword tokenization, and next-sentence prediction objectives, enabling it to learn rich contextual representations of words. With its deep architecture consisting of multiple transformer layers, BERT can capture intricate relationships between words in both left and right contexts, making it highly effective for a wide range of natural language processing tasks such as text classification, named entity recognition, question answering, and more. In our case, we will use the BERT model Camembert-base for token classification which we will fine-tune on our dataset with our tags (presented in the left part of the Figure 5).

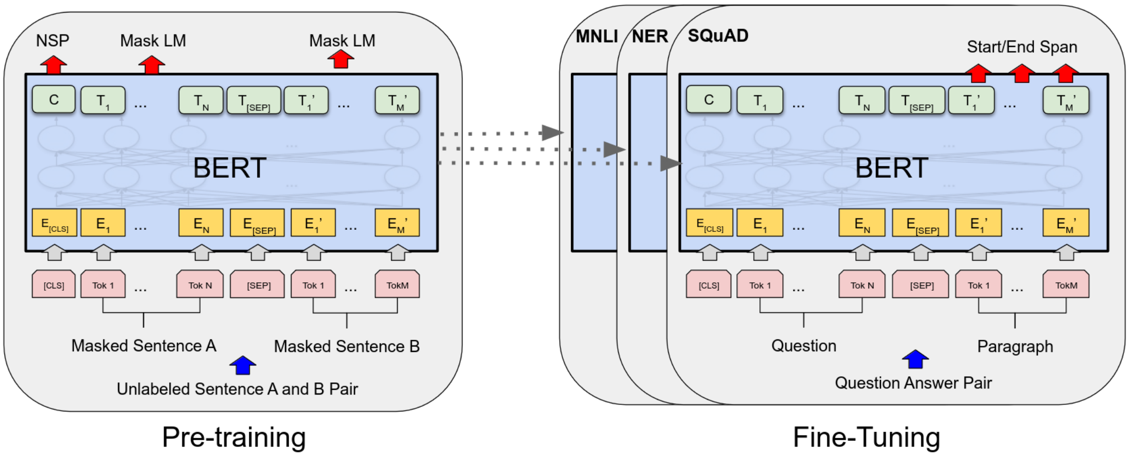


Figure 5: Overall pre-training and fine-tuning procedures for BERT (image from this paper [2])

3.3.2 Chat-GPT

[4] presented a new type of model, which is a Generative Pre-trained Transformer (GPT). It is a type of LLM developed by OpenAI, designed to understand and generate human-like text. It leverages a deep learning architecture based on transformers, which allows it to process and produce coherent and contextually relevant text by predicting subsequent words in a sequence. GPT models are trained on diverse datasets from the internet, enabling them to perform a wide range of NLP tasks such as translation, summarization,

question answering, and text completion or in our case Named Entities recognition. For our work, we use Langchain[7], which has functions that can help the formatting of the response to obtain a specific JSON format.

3.3.3 Gliner

In [11], the authors introduced the GLiNER model. It is trained to extract any types of entity using a Bidirectional Language Models. This model has three main components: a pre-trained textual encoder (a BiLM such as BERT), a span representation module which computes span embeddings from token embeddings, an entity representation module which computes entity embeddings that the model seeks to extract. The goal is to have entity and span embeddings in the same latent space to assess their compatibility (degree of matching).

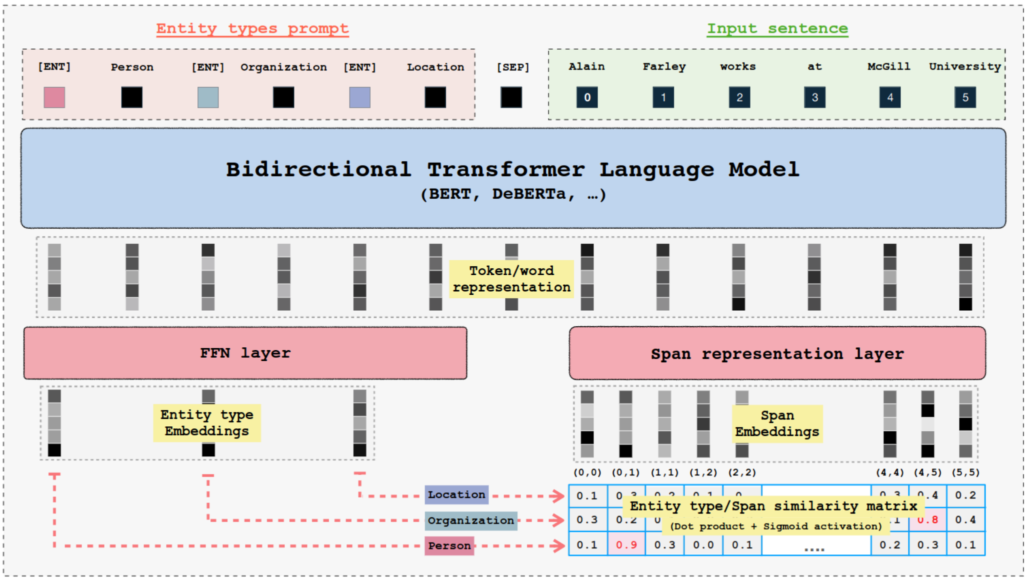


Figure 6: Model architecture (source [11])

3.4 Formats IOB2/IO

When studying NER, we need to deal with format tagging at the token level. There are multiples types but in this work, we have studied two types of format:

1. IO (Inside Outside) format as his name entailed signified that when a token belongs to a named entity we tag it with a specific label or "O" otherwise (such as in the Figure 1 in the column IO where for the entity "petite ville" is labelled with the tag "NC-Spatial").

2. IOB2 (Inside Outside Beginning) format is the same as the IO format except that for the first token composing an entity we add the prefix "B-" and after that we add the prefix "I-". For example in Table 1 in the column IOB2 with the entity "la nouvelle castille" where we have on the first token "la" NP-Spatial becomes " B-NP-Spatial" and the rest of the entity "nouvelle castille" is "I-NP-Spatial".

[7] https://python.langchain.com/v0.1/docs/get_started/introduction/

Token	IO	IOB2	Token	IO	IOB2
ILLESCAS	Head	B-Head	la	NP-Spatial	B-NP-Spatial
,	O	O	nouvelle	NP-Spatial	I-NP-Spatial
(	O	O	castille	NP-Spatial	I-NP-Spatial
Géog	Domain-mark	B-Domain-mark	,	Relation	B-Relation
.	Domain-mark	I-Domain-mark	à	Relation	I-Relation
)	O	O	six	Relation	I-Relation
petite	NC-Spatial	B-NC-Spatial	lieues	Relation	I-Relation
ville	NC-Spatial	I-NC-Spatial	au	Relation	I-Relation
d'	O	O	sud	Relation	I-Relation
Espagne	NP-Spatial	B-NP-Spatial	de	Relation	I-Relation
,	O	O	madrid	NP-Spatial	B-NP-Spatial
dans	Relation	B-Relation	.	O	O

Table 1: Sample of annotations in IO/IOB2 format

4 Methodology

4.1 Dataset Description

Our work uses the GeoEDdA gold standard dataset[8] developed by the team members[9]. This dataset contains labeled data for named entity recognition and span categorization annotations from Diderot & d'Alembert's Encyclopédie entries. The tagset is as follows:

1. NC-Spatial: a common noun that identifies a spatial entity (nominal spatial entity) including natural features, e.g. ville, la rivière, royaume.

2. NP-Spatial: a proper noun identifying the name of a place (spatial named entities), e.g. France, Paris, la Chine.

3. ENE-Spatial: nested spatial entity , e.g. ville de France, royaume de Naples, la mer Baltique.

4. Relation: spatial relation, e.g. dans, sur, à 10 lieues de.

5. Latlong: geographic coordinates, e.g. Long. 19. 49. lat. 43. 55. 44.

6. NC-Person: a common noun that identifies a person (nominal person entity), e.g. roi, l'empereur, les auteurs.

7. NP-Person: a proper noun identifying the name of a person (person named entities), e.g. Louis XIV, Pline, les Romains.

8. ENE-Person: nested people entity, e.g. le czar Pierre, roi de Macédoine

9. NP-Misc: a proper noun identifying entities not classified as spatial or person, e.g. l'Eglise, 1702, Pélasgique.

10. ENE-Misc: nested named entity not classified as spatial or person, e.g. l'ordre de S. Jacques, la déclaration du 21 Mars 1671.

[8] https://huggingface.co/datasets/GEODE/GeoEDdA

11. Head: entry name

12. Domain-Mark: words indicating the knowledge domain (usually after the head and between parenthesis), e.g. Géographie, Geog., en Anatomie

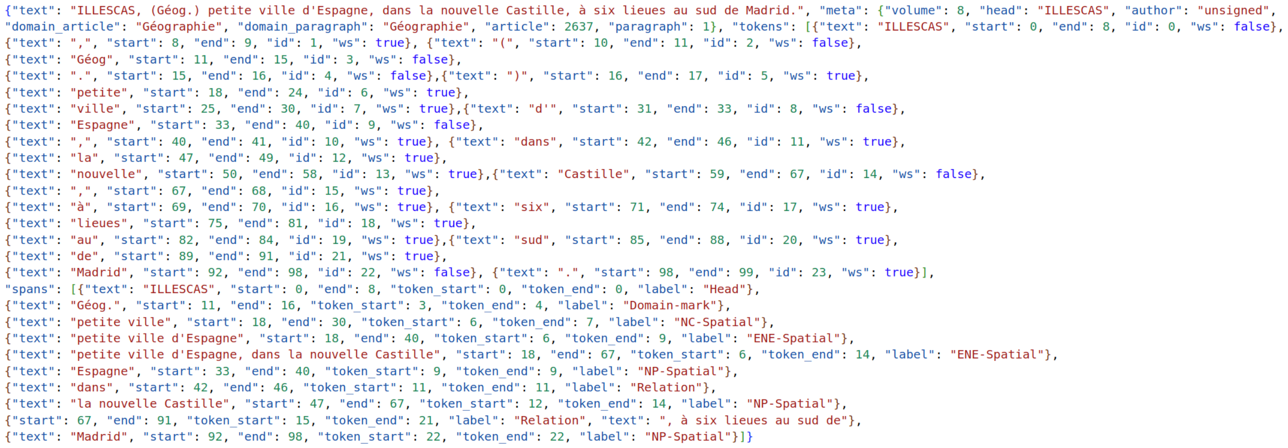
The GeoEDdA dataset comprises 2,200 paragraphs randomly selected from 2,001 entries in the Encyclopédie. Each paragraph is characterized in one category describing the content of it, the different categories are Géographie (1,096 paragraphs), Histoire (259 paragraphs), Droit Jurisprudence (113 paragraphs), Physique (92 paragraphs), Métiers (92 paragraphs), Médecine (88 paragraphs), Philosophie (69 paragraphs), Histoire naturelle (65 paragraphs), Belles-lettres (65 paragraphs), Militaire (62 paragraphs),Commerce (48 paragraphs), Beaux-arts (44 paragraphs), Agriculture (36 paragraphs), Chasse (31 paragraphs), Religion (23 paragraphs), and Musique (17 paragraphs).

We observed that Géographie paragraphs have in average more spatial entities than the average paragraphs, the same thing happen with the History paragraphs with the Person entities. The project team labeled the spans/entities, employing pre-labeling with initial models to expedite the process. They divided the data into training, validation, and test sets. Each validation and test set consists of 200 paragraphs: 100 categorized as "Géographie" and 100 from a different knowledge domain. Table 2 shows the distribution of entities. We observed that there is a variation between the different entities such as between Spatial, Person and Misc.

	Train	*Validation*	*Test*
Paragraphs	1,8	200	200
Tokens	132,398	14,959	13,881
NC-Spatial	3,252	358	355
NP-Spatial	4,707	464	519
ENE-Spatial	3,033	326	334
Relation	2,093	219	226
Latlong	553	66	72
NC-Person	1,378	132	133
NP-Person	1,599	170	150
ENE-Person	492	49	57
NP-Misc	948	108	96
ENE-Misc	255	31	22
Head	1,261	142	153
Domain-Mark	1,069	122	133

Table 2: Distribution of entity across the different sets

The dataset is provided as JSONLines format files, one for each set (e.g., train, validation, test). Each line of the JSONL file contains data for one paragraph. It contains the original text, the category (*Géographie*, *Histoire*, ...), the author, the volume, a list of a dictionary describing the tokens composing the text with additional information about the number describing when the token starts and ends, and the spans of the text with the label, the number of the token where it starts and ends (such as in the Figure 7).



```
{"text": "ILLESCAS, (Géog.) petite ville d'Espagne, dans la nouvelle Castille, à six lieues au sud de Madrid.", "meta": {"volume": 8, "head": "ILLESCAS", "author": "unsigned",
"domain_article": "Géographie", "domain_paragraph": "Géographie", "article": 2637, "paragraph": 1}, "tokens": [{"text": "ILLESCAS", "start": 0, "end": 8, "id": 0, "ws": false},
{"text": ",", "start": 8, "end": 9, "id": 1, "ws": true}, {"text": "(", "start": 10, "end": 11, "id": 2, "ws": false},
{"text": "Géog", "start": 11, "end": 15, "id": 3, "ws": false},
{"text": ".", "start": 15, "end": 16, "id": 4, "ws": false},{"text": ")", "start": 16, "end": 17, "id": 5, "ws": true},
{"text": "petite", "start": 18, "end": 24, "id": 6, "ws": true},
{"text": "ville", "start": 25, "end": 30, "id": 7, "ws": true},{"text": "d'", "start": 31, "end": 33, "id": 8, "ws": false},
{"text": "Espagne", "start": 33, "end": 40, "id": 9, "ws": false},
{"text": ",", "start": 40, "end": 41, "id": 10, "ws": true}, {"text": "dans", "start": 42, "end": 46, "id": 11, "ws": true},
{"text": "la", "start": 47, "end": 49, "id": 12, "ws": true},
{"text": "nouvelle", "start": 50, "end": 58, "id": 13, "ws": true},{"text": "Castille", "start": 59, "end": 67, "id": 14, "ws": false},
{"text": ",", "start": 67, "end": 68, "id": 15, "ws": true},
{"text": "à", "start": 69, "end": 70, "id": 16, "ws": true}, {"text": "six", "start": 71, "end": 74, "id": 17, "ws": true},
{"text": "lieues", "start": 75, "end": 81, "id": 18, "ws": true},
{"text": "au", "start": 82, "end": 84, "id": 19, "ws": true},{"text": "sud", "start": 85, "end": 88, "id": 20, "ws": true},
{"text": "de", "start": 89, "end": 91, "id": 21, "ws": true},
{"text": "Madrid", "start": 92, "end": 98, "id": 22, "ws": false}, {"text": ".", "start": 98, "end": 99, "id": 23, "ws": true}],
"spans": [{"text": "ILLESCAS", "start": 0, "end": 8, "token_start": 0, "token_end": 0, "label": "Head"},
{"text": "Géog.", "start": 11, "end": 16, "token_start": 3, "token_end": 4, "label": "Domain-mark"},
{"text": "petite ville", "start": 18, "end": 30, "token_start": 6, "token_end": 7, "label": "NC-Spatial"},
{"text": "petite ville d'Espagne", "start": 18, "end": 40, "token_start": 6, "token_end": 9, "label": "ENE-Spatial"},
{"text": "petite ville d'Espagne, dans la nouvelle Castille", "start": 18, "end": 67, "token_start": 6, "token_end": 14, "label": "ENE-Spatial"},
{"text": "Espagne", "start": 33, "end": 40, "token_start": 9, "token_end": 9, "label": "NP-Spatial"},
{"text": "dans", "start": 42, "end": 46, "token_start": 11, "token_end": 11, "label": "Relation"},
{"text": "la nouvelle Castille", "start": 47, "end": 67, "token_start": 12, "token_end": 14, "label": "NP-Spatial"},
{"start": 67, "end": 91, "token_start": 15, "token_end": 21, "label": "Relation", "text": ", à six lieues au sud de"},
{"text": "Madrid", "start": 92, "end": 98, "token_start": 22, "token_end": 22, "label": "NP-Spatial"}]}
```

Figure 7: Example of a paragraph

When looking at the different type of entities from the dataset, we found that some entities are only found in the training and testing set and not in the validation set such as ENE-Spatial-4 which complicates the apprenticing of the model. Some others are only present in the training set, such as ENE-Misc-2,ENE-Person-2, ENE-Spatial-5, and ENE-Spatial-6 which makes it hard to evaluate the capacity of our model to find them.

4.2 Multi-label span classification

In this work, the adopted annotation schema contains nested entities [3]. This means that some tokens can belong to several spans and thus have several labels. This overlapping annotations are usually avoided by traditional NER tools. To enable the utilization of a single model for token classification in multi-label scenarios, the team members had previously trained and evaluated a spaCy spancat model[9] [9].

In this work, we investigate the use and compare the results of several architectures such as CRF, CNN, Bi-LSTM, Transformers and LLMs (such as GPT).

We also experimented a "JointLabel" method, as described in [10]. This approach involves amalgamating labels when a token is nested within multiple entities. For instance, if the token "France" belongs to both NP-Spatial and ENE-Spatial spans, the designated label becomes "NP-Spatial+ENE-Spatial

5 Experiments

Throughout our experimentation process, we meticulously explored various setups for evaluating NER systems. Initially, we conducted evaluations using straightforward entities categorized as level 0, excluding any nested entities such as ENE-Person or ENE-Spatial. Subsequently, we extended our evaluation to include level 0 entities along with their nested counterparts. Finally, we assessed the system's capability to identify entities and nested entities across all levels comprehensively. We want to emphasize that while we won't present the results for every individual setup, we will, however, showcase the outcomes of one type of each model that underwent fine-tuning or training.

The different settings are as follows:

1. CRF: We aimed to identify which features can be used to recognize entities, so we trained a CRF exclusively on the named entities (nested or overlapping spans are not included).

[9] https://huggingface.co/GEODE/fr_spacy_custom_spancat_edda

2. CNN Model: We decided to train the JointLabel CNN Model for 50 epochs to identify all named entities and nested entities at all levels.

3. Bi-LSTM Model: To compare two different models using JointLabel, we trained the Flair model for 30 epochs on all named entities and nested entities at all levels.

4. SPAN BERT Model: We opted to train the multi-label BERT Model for 30 epochs to identify all named entities and nested entities at all levels.

5.1 CRF Model

To train a CRF model, we need to first tune two hyperparameters C1 and C2 which represent the regularization constants. C1 controls the L1 regularization, which encourages sparsity in the feature weights by penalizing the absolute values of the weights. L1 regularization can lead to some weights being exactly zero, effectively performing feature selection. C2 controls the L2 regularization, which penalizes the squared values of the weights. L2 regularization tends to spread out the weights more evenly, preventing any single weight from becoming too large. So, we decided to do a GridSearch with a certain set of values [0.5, 0.1, 0.01, 0.001, 0.0001] for C1 and C2 and we found that the best value is 0.5, and 0.1 for C1, and C2 respectively. Our experiment involves training three distinct models, each characterized by unique attributes. The initial model encompasses all parameters except POS (e.g. Part-of-Speech) and DEP (e.g., dependency parsing), while the second includes POS alongside the other parameters. Finally, the third model incorporates all attributes. By employing these configurations and models, we aim to discern the impact (positive and negative) of each parameter of the current token on the labeling process.

	base	base+POS	base+POS+DEP	Support
Domain-mark	98.2	98.6	99.0	392
Head	87.1	87.9	87.7	254
NC-Person	60.9	61.7	66.0	225
NC-Spatial	90.9	91.3	89.4	592
NP-Misc	60.3	64.1	63.8	175
NP-Person	75.9	75.3	77.4	203
NP-Spatial	89.8	90.2	91.1	718
Relation	92.8	92.7	91.0	452
Micro Avg	87.1	87.6	87.4	3011
Macro Avg	82.0	82.7	83.2	3011
Weighted Avg	86.5	87.0	87.1	3011

Table 3: CRF F-scores on test set with different sets of parameter

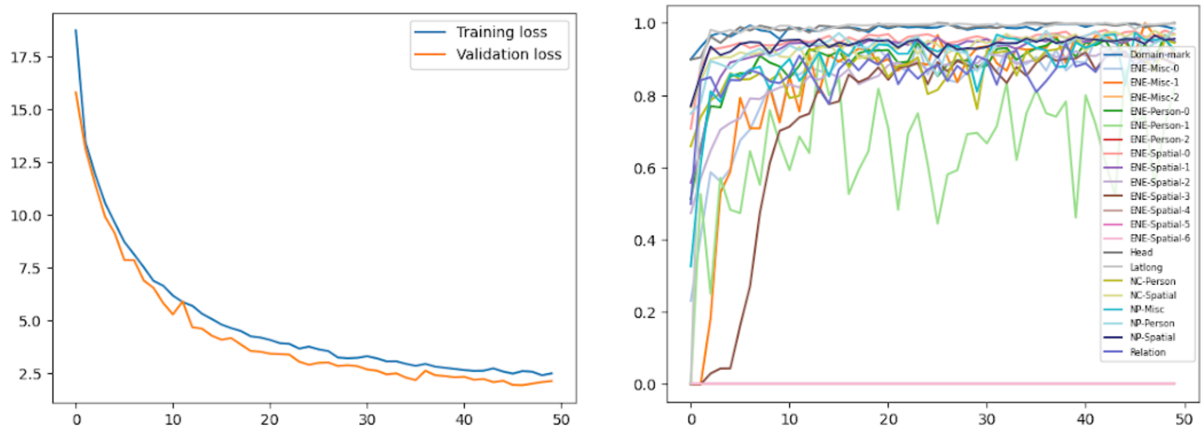
Table 3 shows the F-scores for the three trained CRF models. We observe only small differences (less than 0.5%) between the different sets. For all sets we have good results with all the entities except NP-MISC and NC-Person. Table 4 shows the best and worst features for the model with the base, POS, and DEP parameters that determines the class of the token. For example, for the class 'NC-Person' two best features are '*pape*', and '*roi*' which are two titles describing a person. Or 'NP-Spatial' (i.e., place names) has two of the best features being '*Italie*'.

	Domain mark	*Head*	*Relation*	*NC-Person*	*NC-Spatial*	*NP-Misc*	*NP-Person*	*NP-Spatial*
Top 5 best Features	'Token:.'	'prev_Token:*'	'lower:midi'	'lower:pape'	'lower:royaume'	'shape:dddd'	'prev_Token:Hunauld'	'Token:ltalie'
	'lower:.'	'prev_lower:*'	'Token:dessous'	'lower:roi'	'lower:fleuve'	'Token:persan'	'prev_lower:hunauld'	'lower:ltalie'
	'Token:terme'	'Token:)'	'lower:dessous'	'Token:Mans'	'lower:comté'	'lower:persan'	'lower:juifs'	'prev_lower:palus'
	'lower:terme'	'lower:)'	'Token:Midi'	'lower:mans'	'Token:île'	'prev_Token:xiij'	'Token:bazanés'	'prev_Token:palus'
	'Token:Géograph'	'isupper'	'next_Token:petits'	'lower:président'	'lower:île'	'prev_lower:xiij'	'lower:bazanés'	'lower:indes'
Top 5 worst Features	'shape:Xxxxxxxxxxx'	'prev_lower:)'	'next_lower:se'	'shape:Xxxxxxxxxx'	'next_dep:obl:agent'	'prev_Token:v.'	'prev_shape:Xxxxxxxx'	'shape:xxx'
	'dep:appos'	'prev_shape:X.'	'next_Token:sur'	'isstop'	'Token:du'	'next_shape:dddd'	'shape:xxxxxx'	'next_dep:flat:name'
	'pos:DET'	'prev_pos:NOUN'	'next_lower:sur'	'pos:ADV'	'lower:du'	'pos:ADP'	'shape:xxxx'	'shape:xxxxxxxx'
	'pos:PUNCT'	'prev_pos:PUNCT'	'shape:x.'	'shape:Xxxxx'	'next_dep:det'	'shape:dd'	'shape:xxxxxxxxxxx'	'shape:xxxxxxx'
	'isupper'	'prev_shape:X'	'pos:PROPN'	'next_dep:xcomp'	'isstop'	'isstop'	'shape:xxxxxxxxxx'	'shape:xxxxxxxxxxx'

Table 4: Top 5 best and worst features of CRF Model (Base+Pos+Dep parameters)

5.2 CNN Model

Using the spaCy package, we train the CNN model for up to 50 epochs. The training will stop early if the training loss exceeds the validation loss for 5 consecutive epochs. As shown in Figure 8a, the average training loss consistently surpasses the average validation loss. In Figure 8b, most entities achieve a high F1-score, even if for some classes it has stagnates before increasing during more epochs than other such as ENE-Misc-1, ENE-Spatial-3, and ENE-Person-1. But other entities completely stagnates at 0.0 for example the classes ENE-Spatial-4, ENE-Spatial-5, and ENE-Spatial-6, it is mostly due to the under-representation of these classes in the validation set.



(a)

(b)

Figure 8: (a) Validation and Training Loss for 50 epochs, (b) F1-score on the validation set per entities with a learning rate 0.001

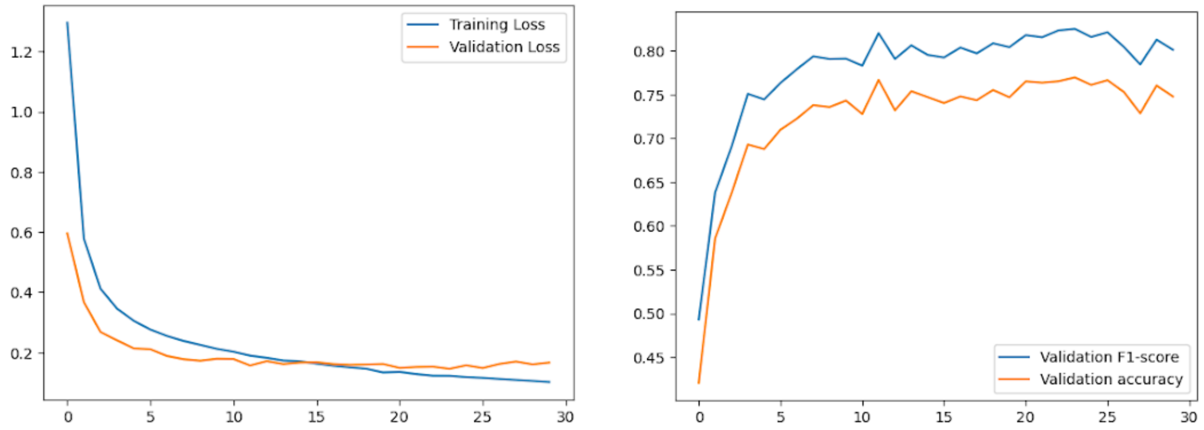
Examining the test set results, we observe high scores for most named entities, except for the *-MISC entities. Similarly, for nested entities, EN-Spatial and ENE-Person show the best performance. However, the performance deteriorates for nested entities at level 1 and higher compared to those at the lower level. These is explain by the low count of nested entities at each level for example at level 0, we have 822 tokens ENE-Spatial-0 and only 40 token for ENE-Spatial-4.

	Precision	*Recall*	*F-score*	*Support*		*Precision*	*Recall*	*F-score*	*Support*
Domain-mark	90.6	98.0	94.1	392	*ENE-Misc-2*	0.00	0.00	0.00	0
Head	91.2	85.8	88.4	254	*ENE-Person-0*	81.5	53.3	64.4	199
NC-Person	56.7	67.6	61.7	225	*ENE-Person-1*	50.0	4.8	8.7	21
NC-Spatial	91.4	83.1	87.1	592	*ENE-Person-2*	0.00	0.00	0.00	0
NP-Misc	43.3	66.3	52.4	175	*ENE-Spatial-0*	87.9	82.4	85.1	802
NP-Person	69.1	78.3	73.4	203	*ENE-Spatial-1*	77.3	61.8	68.7	685
NP-Spatial	90.1	77.0	83.0	718	*ENE-Spatial-2*	43.5	57.2	49.4	425
Relation	89.7	71.2	79.4	452	*ENE-Spatial-3*	24.2	9.1	13.3	175
Latlong	95.9	94.8	95.3	789	*ENE-Spatial-4*	21.7	12.5	15.9	40
ENE-Misc-0	27.1	28.4	27.7	81	*ENE-Spatial-5*	0.00	0.00	0.00	0
ENE-Misc-1	0.00	0.00	0.00	5	*ENE-Spatial-6*	0.00	0.00	0.00	0
Micro Avg	78.5	74.2	76.3	6223					
Macro Avg	51.4	46.9	47.6	6233					
Weighted Avg	79.4	74.2	76.1	6233					

Table 5: CNN F1-score on the test set per entities

5.3 Bi-LSTM Model

When training a BI-LSTM, we use a learning rate of 0.200 but it will decrease by 10% if during 5 epochs consecutive the f1-score on the validation set don't increase. In Figure 9a, we have the validation loss being superior to the training loss only after 15 epochs and when looking at the learning rate evolution, it decreases two times at the 23rd epoch, and at the 29th epoch. When looking at the Figure 9b, we have f1-score and accuracy increasing during the first 5 epochs, then they stagnate at 0.80 for the rest of the training.



(a)

(b)

Figure 9: (a) Validation and Training Loss for 30 epochs, (b) F1-score and accuracy on the validation set

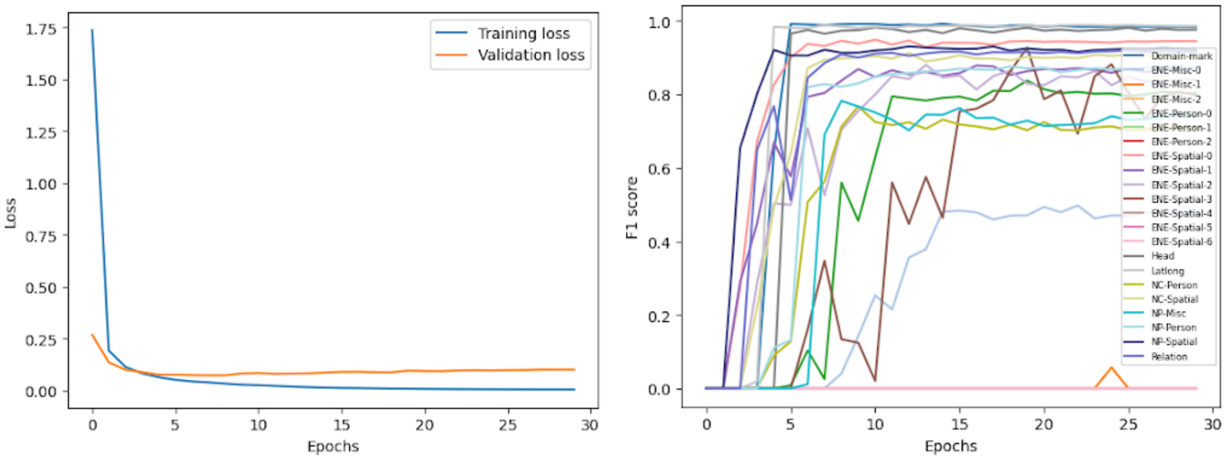
On the test set, we have better results than the CNN model in general (on all the basic named entities), but we encounter the same difficulty as the previous model with the Misc and Person entities and the nested entities with a level higher than level 1.

	Precision	*Recall*	*F-score*	*Support*		*Precision*	*Recall*	*F-score*	*Support*
Domain-mark	99.2	99.0	99.1	392	*ENE-Misc-2*	0.00	0.00	0.00	0
Head	96.0	94.9	95.4	254	*ENE-Person-0*	90.9	70.4	79.3	199
NC-Person	64.5	87.1	74.1	225	*ENE-Person-1*	100	19.0	32.0	21
NC-Spatial	90.3	95.6	92.9	592	*ENE-Person-2*	0.00	0.00	0.00	0
NP-Misc	73.1	76.0	74.5	175	*ENE-Spatial-0*	93.3	89.9	91.6	802
NP-Person	86.4	90.6	88.5	203	*ENE-Spatial-1*	91.2	83.2	87.0	685
NP-Spatial	95.7	95.3	95.5	718	*ENE-Spatial-2*	76.9	91.1	83.4	425
Relation	93.3	93.1	93.2	452	*ENE-Spatial-3*	69.3	76.0	72.5	175
Latlong	99.2	97.6	98.4	789	*ENE-Spatial-4*	1.8	2.5	2.1	40
ENE-Misc-0	37.0	42.0	39.3	81	*ENE-Spatial-5*	0.00	0.00	0.00	0
ENE-Misc-1	0.00	0.00	0.00	5	*ENE-Spatial-6*	0.00	0.00	0.00	0
Micro Avg	88.3	89.4	88.9	6223					
Macro Avg	61.7	59.2	59.0	6233					
Weighted Avg	89.2	89.4	89.0	6233					

Table 6: Bi-LSTM F1-score on the test set per entities

5.4 Span BERT Model

After modifying the basic BERT-like Camembert model to be able to apply multilabel classification, we train this model for 30 epochs with a learning rate 0.0001, and we only save the model with the best f1-score on the validation set. In the Figure 10a, we observed that after 5 epochs the average training loss is lower than the average validation loss, so after this epoch our model over-fitted. And when looking at the f1-score on the validation set, we have a majority of entities increasing in the first epochs and then stagnating around the 90% f1-score but for the nested entities we need more epochs to detect them and learning to identify these classes. And as with the other models, the nested entities ENE-Spatial-4, ENE-Spatial-5, and ENE-Spatial-6 don't increase.



(a) **(b)**

Figure 10: (a) Validation and Training Loss for 50 epochs, (b) F1-score on the validation set per entities

Examining the test set results, we achieved excellent performance in NER. However, we encountered similar difficulties with the *-MISC category and nested entities beyond the first level. Despite these challenges, this model delivered the best overall results when compared to other models.

	Precision	*Recall*	*F-score*	*Support*		*Precision*	*Recall*	*F-score*	*Support*
Domain-mark	99.7	99.0	99.4	392	*ENE-Misc-2*	0.00	0.00	0.00	0
Head	97.3	98.0	97.6	254	*ENE-Person-0*	88.3	79.9	83.9	199
NC-Person	69.1	85.3	76.3	225	*ENE-Person-1*	0.00	0.00	0.00	21
NC-Spatial	88.4	95.4	91.8	592	*ENE-Person-2*	0.00	0.00	0.00	0
NP-Misc	69.5	79.4	74.1	175	*ENE-Spatial-0*	93.2	92.5	92.9	802
NP-Person	87.6	86.7	87.1	203	*ENE-Spatial-1*	84.7	85.4	85.0	685
NP-Spatial	97.0	94.4	95.7	718	*ENE-Spatial-2*	76.1	94.6	84.4	425
Relation	86.9	95.6	91.0	452	*ENE-Spatial-3*	70.3	88.0	78.2	175
Latlong	96.3	98.1	97.2	789	*ENE-Spatial-4*	0.00	0.00	0.00	40
ENE-Misc-0	35.1	49.4	41.0	81	*ENE-Spatial-5*	0.00	0.00	0.00	0
ENE-Misc-1	0.00	0.00	0.00	5	*ENE-Spatial-6*	0.00	0.00	0.00	0
Micro Avg	87.4	91.0	89.2	6223					
Macro Avg	56.3	60.1	58.0	6233					
Weighted Avg	87.3	91.0	89.0	6233					

Table 7: Span BERT F1-score on the test set per entities

In Table 8, we can find the results for all the models presented before and also the *fr_spacy_custom_spancat_edda* model which is the baseline trained on the Gold standard dataset by the GEODE project team with spaCy to identify spans. In comparison, we can see that our models have better results on entities such as Head, Domain-Mark, Latlong but they are worse than the custom spaCy with the nested entities.

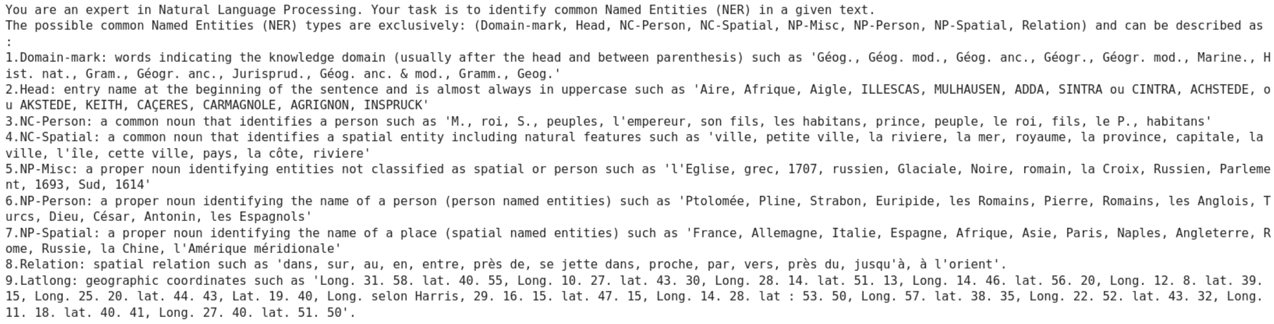
	BASE+POS+DEP FEATURES	*BASE BERT*	*SPAN BERT*	*SPACY SPANCAT*	*BI-LSTM MODEL*	*CNN Model*
	CRF MODEL NER	*WITH ALL TAGS*	*WITH ALL TAGS*		*WITH ALL TAGS*	*WITH ALL TAGS*
Domain-mark	99.0	99.9	99.4	95.8	99.1	94.1
Head	87.7	97.5	97.6	45.1	95.4	88.4
Relation	91.0	91.4	91.0	52.5	93.2	79.4
Latlong	-	97.4	97.2	0.00	98.4	95.3
NC-Person	66.0	73.5	76.3	78.0	74.1	61.7
NC-Spatial	89.4	92.6	91.8	95.3	92.9	87.1
NP-Misc	63.8	71.5	74.1	71.9	74.5	52.4
NP-Person	77.4	87.3	87.1	93.0	88.5	73.4
NP-Spatial	91.1	95.4	95.7	95.4	95.5	83.0
ENE-Misc-0	-	50.5	41.0	0.00	39.3	27.7
ENE-Misc-1	-	0.00	0.00	0.00	0.00	0.00
ENE-Person-0	-	84.0	83.9	88.2	79.3	64.4
ENE-Person-1	-	64.9	0.00	41.0	32.0	8.7
ENE-Spatial-0	-	92.3	92.9	94.1	91.6	85.1
ENE-Spatial-1	-	86.0	85.0	89.6	87.0	68.7
ENE-Spatial-2	-	79.7	84.4	87.8	83.4	49.4
ENE-Spatial-3	-	67.7	78.2	78.0	72.5	13.3
ENE-Spatial-4	-	36.4	0.00	51.6	2.1	15.9

Table 8: Summary of F1-score across the different models

5.5 Chat-GPT

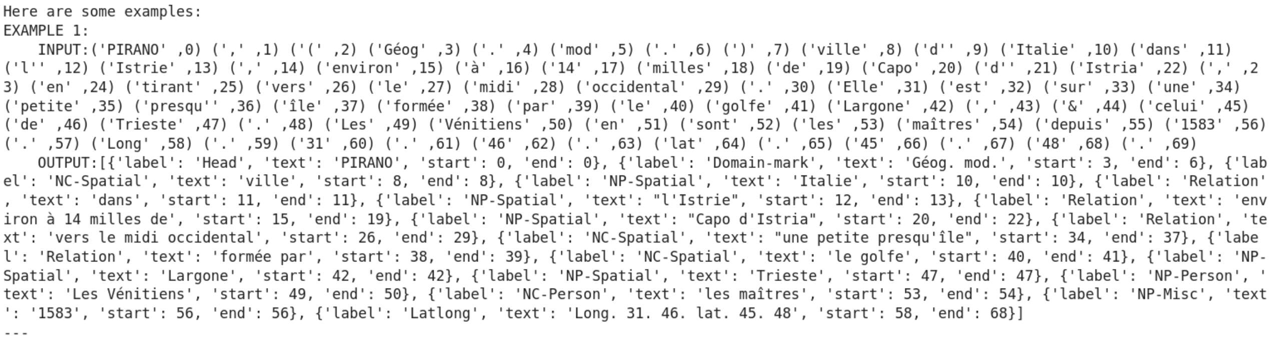
Additionnaly, we also experimented with LLMs and GPT models. In our tests, we use GPT-3.5 with Langchain to identify only the named entities (excluding nested named entities). To achieve that, we had first to create a prompt to give the context to GPT, defining the different entities with examples, and giving examples to help understand what we want (few shot learning). In these examples, we had to format them to have the same entity as in our JSONLines input dataset. In Figure 11, you can see the introduction with the context of the task, the definition of our classes and some examples. In 12, an example for Langchain where we have the input formatted with each token enumerated and in the output the format with 'start', 'end', 'label', and 'text'.

Our experiments showed that if we give the input text of our request without information about the tokens (start and end positions), GPT understands how to format the output but fails filling correct numbers. For this reason, we give the input as an already tokenized text which seems to help the model.



```
You are an expert in Natural Language Processing. Your task is to identify common Named Entities (NER) in a given text.
The possible common Named Entities (NER) types are exclusively: (Domain-mark, Head, NC-Person, NC-Spatial, NP-Misc, NP-Person, NP-Spatial, Relation) and can be described as
:
1.Domain-mark: words indicating the knowledge domain (usually after the head and between parenthesis) such as 'Géog., Géog. mod., Géog. anc., Géogr., Géogr. mod., Marine., H
ist. nat., Gram., Géogr. anc., Jurisprud., Géog. anc. & mod., Gramm., Geog.'
2.Head: entry name at the beginning of the sentence and is almost always in uppercase such as 'Aire, Afrique, Aigle, ILLESCAS, MULHAUSEN, ADDA, SINTRA ou CINTRA, ACHSTEDE, o
u AKSTEDE, KEITH, CAÇERES, CARMAGNOLE, AGRIGNON, INSPRUCK'
3.NC-Person: a common noun that identifies a person such as 'M., roi, S., peuples, l'empereur, son fils, les habitans, prince, peuple, le roi, fils, le P., habitans'
4.NC-Spatial: a common noun that identifies a spatial entity including natural features such as 'ville, petite ville, la riviere, la mer, royaume, la province, capitale, la
ville, l'île, cette ville, pays, la côte, riviere'
5.NP-Misc: a proper noun identifying entities not classified as spatial or person such as 'l'Eglise, grec, 1707, russien, Glaciale, Noire, romain, la Croix, Russien, Parleme
nt, 1693, Sud, 1614'
6.NP-Person: a proper noun identifying the name of a person (person named entities) such as 'Ptolomée, Pline, Strabon, Euripide, les Romains, Pierre, Romains, les Anglois, T
urcs, Dieu, César, Antonin, les Espagnols'
7.NP-Spatial: a proper noun identifying the name of a place (spatial named entities) such as 'France, Allemagne, Italie, Espagne, Afrique, Asie, Paris, Naples, Angleterre, R
ome, Russie, la Chine, l'Amérique méridionale'
8.Relation: spatial relation such as 'dans, sur, au, en, entre, près de, se jette dans, proche, par, vers, près du, jusqu'à, à l'orient'.
9.Latlong: geographic coordinates such as 'Long. 31. 58. lat. 40. 55, Long. 10. 27. lat. 43. 30, Long. 28. 14. lat. 51. 13, Long. 14. 46. lat. 56. 20, Long. 12. 8. lat. 39.
15, Long. 25. 20. lat. 44. 43, Lat. 19. 40, Long. selon Harris, 29. 16. 15. lat. 47. 15, Long. 14. 28. lat : 53. 50, Long. 57. lat. 38. 35, Long. 22. 52. lat. 43. 32, Long.
11. 18. lat. 40. 41, Long. 27. 40. lat. 51. 50'.
```

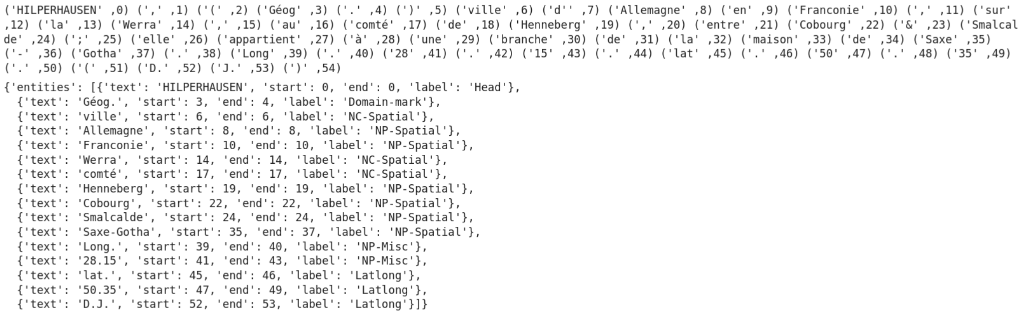
Figure 11: Introduction of our prompt for Chat-GPT-3.5



```
Here are some examples:
EXAMPLE 1:
    INPUT:('PIRANO' ,0) (',' ,1) ('(' ,2) ('Géog' ,3) ('.' ,4) ('mod' ,5) ('.' ,6) (')' ,7) ('ville' ,8) ('d'' ,9) ('Italie' ,10) ('dans' ,11)
('l'' ,12) ('Istrie' ,13) (',' ,14) ('environ' ,15) ('à' ,16) ('14' ,17) ('milles' ,18) ('de' ,19) ('Capo' ,20) ('d'' ,21) ('Istria' ,22) (',' ,2
3) ('en' ,24) ('tirant' ,25) ('vers' ,26) ('le' ,27) ('midi' ,28) ('occidental' ,29) ('.' ,30) ('Elle' ,31) ('est' ,32) ('sur' ,33) ('une' ,34)
('petite' ,35) ('presqu'' ,36) ('île' ,37) ('formée' ,38) ('par' ,39) ('le' ,40) ('golfe' ,41) ('Largone' ,42) (',' ,43) ('&' ,44) ('celui' ,45)
('de' ,46) ('Trieste' ,47) ('.' ,48) ('Les' ,49) ('Vénitiens' ,50) ('en' ,51) ('sont' ,52) ('les' ,53) ('maîtres' ,54) ('depuis' ,55) ('1583' ,56)
('.' ,57) ('Long' ,58) ('.' ,59) ('31' ,60) ('.' ,61) ('46' ,62) ('.' ,63) ('lat' ,64) ('.' ,65) ('45' ,66) ('.' ,67) ('48' ,68) ('.' ,69)
    OUTPUT:[{'label': 'Head', 'text': 'PIRANO', 'start': 0, 'end': 0}, {'label': 'Domain-mark', 'text': 'Géog. mod.', 'start': 3, 'end': 6}, {'lab
el': 'NC-Spatial', 'text': 'ville', 'start': 8, 'end': 8}, {'label': 'NP-Spatial', 'text': 'Italie', 'start': 10, 'end': 10}, {'label': 'Relation'
, 'text': 'dans', 'start': 11, 'end': 11}, {'label': 'NP-Spatial', 'text': "l'Istrie", 'start': 12, 'end': 13}, {'label': 'Relation', 'text': 'env
iron à 14 milles de', 'start': 15, 'end': 19}, {'label': 'NP-Spatial', 'text': "Capo d'Istria", 'start': 20, 'end': 22}, {'label': 'Relation', 'te
xt': 'vers le midi occidental', 'start': 26, 'end': 29}, {'label': 'NC-Spatial', 'text': "une petite presqu'île", 'start': 34, 'end': 37}, {'labe
l': 'Relation', 'text': 'formée par', 'start': 38, 'end': 39}, {'label': 'NC-Spatial', 'text': 'le golfe', 'start': 40, 'end': 41}, {'label': 'NP-
Spatial', 'text': 'Largone', 'start': 42, 'end': 42}, {'label': 'NP-Spatial', 'text': 'Trieste', 'start': 47, 'end': 47}, {'label': 'NP-Person', '
text': 'Les Vénitiens', 'start': 49, 'end': 50}, {'label': 'NC-Person', 'text': 'les maîtres', 'start': 53, 'end': 54}, {'label': 'NP-Misc', 'text
': '1583', 'start': 56, 'end': 56}, {'label': 'Latlong', 'text': 'Long. 31. 46. lat. 45. 48', 'start': 58, 'end': 68}]
---
```

Figure 12: Example for Langchain

The result, it returned was good but it still had difficulties identifying and understanding what represents the start and end parameters of the entity and sometimes it divides one entity into two such as in Figure 13 where we have three entities Latlong consecutive instead of one unique Latlong. Additionnaly, when we give long or multiple examples, it will lose itself and simply return the entities of the examples.



```
('HILPERHAUSEN' ,0) (',' ,1) ('(' ,2) ('Géog' ,3) ('.' ,4) (')' ,5) ('ville' ,6) ('d'' ,7) ('Allemagne' ,8) ('en' ,9) ('Franconie' ,10) (',' ,11) ('sur'
,12) ('la' ,13) ('Werra' ,14) (',' ,15) ('au' ,16) ('comté' ,17) ('de' ,18) ('Henneberg' ,19) (',' ,20) ('entre' ,21) ('Cobourg' ,22) ('&' ,23) ('Smalcal
de' ,24) (';' ,25) ('elle' ,26) ('appartient' ,27) ('à' ,28) ('une' ,29) ('branche' ,30) ('de' ,31) ('la' ,32) ('maison' ,33) ('de' ,34) ('Saxe' ,35)
('-' ,36) ('Gotha' ,37) ('.' ,38) ('Long' ,39) ('.' ,40) ('28' ,41) ('.' ,42) ('15' ,43) ('.' ,44) ('lat' ,45) ('.' ,46) ('50' ,47) ('.' ,48) ('35' ,49)
('.' ,50) ('(' ,51) ('D.' ,52) ('J.' ,53) (')' ,54)
{'entities': [{'text': 'HILPERHAUSEN', 'start': 0, 'end': 0, 'label': 'Head'},
  {'text': 'Géog.', 'start': 3, 'end': 4, 'label': 'Domain-mark'},
  {'text': 'ville', 'start': 6, 'end': 6, 'label': 'NC-Spatial'},
  {'text': 'Allemagne', 'start': 8, 'end': 8, 'label': 'NP-Spatial'},
  {'text': 'Franconie', 'start': 10, 'end': 10, 'label': 'NP-Spatial'},
  {'text': 'Werra', 'start': 14, 'end': 14, 'label': 'NC-Spatial'},
  {'text': 'comté', 'start': 17, 'end': 17, 'label': 'NC-Spatial'},
  {'text': 'Henneberg', 'start': 19, 'end': 19, 'label': 'NP-Spatial'},
  {'text': 'Cobourg', 'start': 22, 'end': 22, 'label': 'NP-Spatial'},
  {'text': 'Smalcalde', 'start': 24, 'end': 24, 'label': 'NP-Spatial'},
  {'text': 'Saxe-Gotha', 'start': 35, 'end': 37, 'label': 'NP-Spatial'},
  {'text': 'Long.', 'start': 39, 'end': 40, 'label': 'NP-Misc'},
  {'text': '28.15', 'start': 41, 'end': 43, 'label': 'NP-Misc'},
  {'text': 'lat.', 'start': 45, 'end': 46, 'label': 'Latlong'},
  {'text': '50.35', 'start': 47, 'end': 49, 'label': 'Latlong'},
  {'text': 'D.J.', 'start': 52, 'end': 53, 'label': 'Latlong'}]}
```

Figure 13: Example of result with GPT using Langchain

We also use other different LLMs in different languages with LM Studio[10] such as Phi3 (created by Microsoft), Gemma (by Google), Mistral (created by a MistralAI), Qwen (created by the Qwen Team, affiliated with Alibaba Group),or Llama (by Meta) to compare them with GPT. We had very varying results, some LLMs provide the format of the entities but don't really understand the labels we want, and sometimes they create new labels. Others completely don't understand the task and do anything or simply repeat the sentence we give to it.

[10] https://lmstudio.ai

5.6 Hybrid model

The results are not equivalent across all classes, with lower scores on the nested entities. We decided to study hybrid models to introduce greater robustness. Hybrid models combine neural and symbolic approaches in order to benefit from the advantages of both types of approach: reasoning on a large amount of data and integrating expert knowledge by means of rules. This hybrid model will consist of our best NER model, which in this case is the BERT model, combined with grammatical rules to detect nested entities. Specifically, the nested entities at level 0 will always be composed of a nominal entity (NC) followed potentially by a word that has part-of-speech function as *ADP* or *ADJ* or *DET* followed by a named entity (NP), and the type of nested entity will always be determined by the nominal entity. For those of higher levels, it will be the same structure except the nominal entity, and the named entity can be replaced by the nested entity. The first preliminary results do not seem entirely satisfactory at this stage of development, but merit further investigation.

6 Conclusion

In conclusion, with the evaluation of the different models to identify the entities done, we can observe that the models with better results than the spacy spancat are the SPAN BERT or the BASE BERT Joint-Label and the Bi-LSTM Model. These results can be explained by the special structure of these two models using embedding for the word or subword. When using the GPT, Langchain has given us good results with the introduction, formatting, and entities in one-shot learning but it still has problems with long examples or multiple examples. But with the other LLMs from LM studio, we got varying results from good format but with unknown labels and not understanding the signification of 'start' and 'end' to a complete incomprehension of the task demanded or a endless repeat of one sentence.

7 Reflection and perspectives

After the last 5 months, I have been to learn a lot in NLP and specifically the NER task, how to use the different packages, and different architecture such as Transformers and also how to apply LLMs such as GPT and how we evaluate them. I have been able to apply the knowledge I learn in this year of the Master 2 Machine Learning Data Mining such as the use of the transformers models, the creation of a deep learning network. Also, I will continue to work on this project until the end of July on multiple objectives:

- modify our GeoEDdA dataset to have a better balance between the different classes among sets (train, validation, and test)
- research other architectures for the detection of named entities recognition
- clean the code and release it on Github

8 Acknowledgements

I would like to express my gratitude to my internship supervisor at LIRIS, Ludovic Moncla, for his dedicated guidance and supervision. His regular weekly meetings to monitor my progress were invaluable. I am also thankful for his role in the interview and selection process, and for the opportunity to work at LIRIS. I am grateful to Yassir Lairgi, a PhD student, for assisting me with the Langchain package and explaining the format for using Chat-GPT. Lastly, I would like to thank Jean Monnet University for allowing me to undertake this internship as part of my Master's program, enabling me to gain experience in an external company environment.

9 References

[1] Rayner Alfred et al. “Malay named entity recognition based on rule-based approach”. In: (2014).

[2] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[3] Mauro Gaio and Ludovic Moncla. “Extended named entity recognition using finite-state transducers: An application to place names”. In: *The ninth international conference on advanced geographic information systems, applications, and services (GEOProcessing 2017)*. 2017.

[4] Katikapalli Kalyan. “A survey of GPT-3 family large language models including ChatGPT and GPT-4”. In: *Natural Language Processing Journal* 6 (Dec. 2023), p. 100048. DOI: 10.1016/j.nlp.2023.100048.

[5] Michal Konkol and Miloslav Konopík. “CRF-based Czech named entity recognizer and consolidation of Czech NER research”. In: *Text, Speech, and Dialogue: 16th International Conference, TSD 2013, Pilsen, Czech Republic, September 1-5, 2013. Proceedings 16*. Springer. 2013, pp. 153–160.

[6] *TextMine’24*. Jan. 2024. URL: https://cnrs.hal.science/hal-04419844.

[7] Alice Millour et al. “Unveiling Strengths and Weaknesses of NLP Systems Based on a Rich Evaluation Corpus: the Case of NER in French”. In: *LREC-COLING 2024*. 2024.

[8] Ludovic Moncla and Mauro Gaio. “Perdido: Python library for geoparsing and geocoding French texts”. In: *First International Workshop on Geographic Information Extraction from Texts (GeoExT)*. Dublin, Ireland, Apr. 2023. URL: https://hal.science/hal-04049794.

[9] Ludovic Moncla, Denis Vigier, and Katherine Mcdonough. “GeoEDdA: A Gold Standard Dataset for Geo-semantic Annotation of Diderot & d’Alembert’s Encyclopédie”. In: *Second International Workshop on Geographic Information Extraction from Texts (GeoExT) to be held at the 46th European Conference on Information Retrieval (ECIR 2024)*. Glasgow, United Kingdom, Mar. 2024. URL: https://hal.science/hal-04511909.

[10] Solenn Tual et al. *A Benchmark of Nested Named Entity Recognition Approaches in Historical Structured Documents*. 2023. arXiv: 2302.10204 [cs.IR].

[11] Urchade Zaratiana et al. *GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer*. 2023. arXiv: 2311.08526 [cs.CL].

[12] Zhehuan Zhao et al. “ML-CNN: A novel deep learning based disease named entity recognition architecture”. In: *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2016, pp. 794–794. DOI: 10.1109/BIBM.2016.7822625.