

UNIVERSITÉ DE LORRAINE - IDMC

MASTER THESIS

Discourse Marker Identification in French Spoken Corpora: Using Rule-Based Method and Machine Learning

Author:
Abdelhalim Hafedh Dahou

Supervisors:
Mathilde Dargnat,
Jacques Jayez,
Mathieu Constant.

Host Lab:
ATILF

A report submitted in fulfillment of the requirements for the degree of Msc in Natural Language Processing

in the

Institut des Sciences du Digital, Management & Cognition

August 24, 2021

Declaration of Authorship

I, Abdelhalim Hafedh Dahou, declare that this report titled, "Discourse Marker Identification in French Spoken Corpora: Using Rule-Based Method and Machine Learning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the report is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:
Abdelhalim hafedh DAHOU

Date:
24 - 08 - 2021

UNIVERSITÉ DE LORRAINE

Abstract

Institut des Sciences du Digital, Management & Cognition

MSc of Natural Language Processing

Discourse Marker Identification in French Spoken Corpora: Using Rule-Based Method and Machine Learning

by Abdelhalim hafedh DAHOU

The objective of this report is to study the identification of French discourse markers (DM), in particular the *polyfunctional* occurrences. A number of words identified as DM, and traditionally considered as adverbs or interjections, are also, for instance, adjectives or nouns. For example *bon* can be a DM or an adjective, *attention* can be a DM or a noun, etc. Hand annotation is in general robust but time consuming. The main difficulty with automatic identification is to take the context of the DM candidate correctly into account.

To do that, a mechanism based on rule-based and machine learning approaches was built, in order to reach an acceptable level of performance and reduce the expert effort.

This report will provide a comprehensive use case of a machine learning algorithm which has proved a good efficiency in dealing with such linguistic phenomena. In addition, an evaluation was done for the Unitex platform in order to determine the efficiency and drawbacks of this platform when dealing with such types of tasks.

Acknowledgements

It is customary to say that a dissertation is not the fruit of the sole work of its author, but the result of numerous and close collaborations; it does not deviate from the rule. I thank God above all for having given us the will to finish this thesis.

This work was born with a lot of help and encouragement from people around us.This short thanks will not be enough to reward their efforts.

All my thoughts of gratitude go to my two thesis supervisors, Prof. Dr. Mathilde DARGNAT, Researcher at ATILF (UMR 7118, CNRS & University of Lorraine), and, Prof. Dr. Mathieu CONSTANT, Researcher at ATLIF(UMR 7118 - CNRS / University of Lorraine), for their steadfast support, unrelenting encouragement, and intelligent advice throughout the thesis process. Their guidance from the beginning to the end allowed me to have a better comprehension of the subject and finish the job. This thesis would not have been possible without their extension expertise and unwavering passion.

I'd also like to express my gratitude to Prof. Dr. Jacques JAYEZ Researcher at ENS Lyon & ISC Marc Jeannerod, UMR 5304 CNRS, for his constant direction, assistance, and various practical advice on the techniques implementation and how to better present this thesis.

I would also like to thank Prof. Dr. Maxime AMBLARD, Head of MSc in NLP and Researcher at SEMAGRAMME team of LORIA ,and also , Prof. Dr. Miguel COUCEIRO, Responsible for 2nd year of Master and Researcher at ORPAILLEUR team of LORIA, for giving me the opportunity to ameliorate my career in the IDMC and being so understanding.

I would also like to thank the members of the jury who agreed to examine us. I send my sincerest thanks to all the colleagues and friends who share with me the good moments of joy during these year.

A Huge thanks to my parents for helping me get here and a special thanks to my lovely soulmate El-yakout. Hope I'll always make them proud. Also , I express my gratitude to all my loved ones who have always supported and encouraged me during the realization of this work. Finally, all those who have contributed, from near or far to the realization of this memory and which I cannot unfortunately quote, find here the expression of my deep gratitude.

Abdelhalim Hafedh

Contents

Declaration of Authorship iii

Abstract v

Acknowledgements vii

1 Introduction 1
- 1.1 Main Question . . . 2
- 1.2 Host Organization . . . 2
- 1.3 Structure of the Report . . . 3

2 State Of The Field 5
- 2.1 Discourse and discourse markers . . . 5
- 2.2 Previous Related Work . . . 5
- 2.3 Resources used . . . 6
  - 2.3.1 Unitex platform . . . 6
  - 2.3.2 CamemBERT . . . 7
  - 2.3.3 FlauBERT . . . 8

3 Methodology 11
- 3.1 Unitex with internal and external ressources . . . 12
  - 3.1.1 Unitex with internal resources . . . 12
    - Read and preprocess the spoken corpus . . . 13
    - Construct the text automaton . . . 15
    - Build and apply graphs . . . 16
  - 3.1.2 Unitex with external resources . . . 20
    - Unitex reading tagged corpus . . . 20
    - Construct the text automaton . . . 21
    - Build and apply graphs . . . 21
- 3.2 Syntactic and lexical patterns . . . 23
  - 3.2.1 *Le Monde* corpus . . . 25
  - 3.2.2 DM negative context extraction . . . 25
  - 3.2.3 Patterns extractions using CamemBERT . . . 25
  - 3.2.4 Apply the patterns . . . 26
  - 3.2.5 Filter process . . . 26
- 3.3 Fine-tuning Pretrained Bert model for PoS Tagging task . . . 26
  - 3.3.1 Model architecture . . . 27
  - 3.3.2 Training data preprocessing . . . 28
  - 3.3.3 Training configurations . . . 28
- 3.4 KNN algorithm with BERT word embedding . . . 29
  - 3.4.1 Training data preparation . . . 30
  - 3.4.2 Find the K value . . . 31
  - 3.4.3 Extract the word embedding . . . 31

3.4.4 Distance calculation 31
3.4.5 Classification process 32

4 Evaluation and Discussion **33**
4.1 Data 33
4.2 Experiments and results 35
4.2.1 Unitex with internal and external resources 35
Results 36
4.2.2 Syntactic and lexical patterns 36
Results 37
4.2.3 Fine-tuning pretrained Bert model for POS tagging task 37
Results 38
4.2.4 KNN algorithm with BERT word embedding 38
Results 38
4.3 Discussion 38
4.3.1 Rule-based approach methods 39
4.3.2 Machine learning approach methods 40

5 Conclusion **41**

Bibliography **43**

List of Figures

3.1 illustration of the methods used in this study with their variants. . . . 12
3.2 Unitex with internal resources pipeline 13
3.3 Result of applying dictionaries to a text. 14
3.4 DELAF syntax example. 14
3.5 Sentence automaton example. 15
3.6 Text automaton linearized. 16
3.7 Graph of the marker *quoi*. 17
3.8 Graph of the marker *attention*. 18
3.9 Graph of the marker *bon*. 18
3.10 Graph of the marker *la preuve*. 19
3.11 Concordance obtained with grammar. 19
3.12 Unitex with external resources pipeline 20
3.13 Tagged text with CamemBERT. 21
3.14 Graph of the marker *Quoi* with CamemBERT tags. 22
3.15 Graph of the marker *Attention* with CamemBERT tags. 22
3.16 Graph of the marker *bon* with CamemBERT tags. 23
3.17 Graph of the marker *La preuve* with CamemBERT tags. 23
3.18 Syntactic and lexical patterns pipeline process. 24
3.19 The pattern extraction results 25
3.20 Fine-tuned Model architecture. 27
3.21 The process of preparing the data used by the pre-trained model. . . . 28
3.22 KNN algorithm using word embedding pipeline.. 30
3.23 The structure of training data on the system. 31
3.24 Cosine similarity equation and illustration. 32

4.1 deployment of the DM in the CORPAIX corpus. 33
4.2 deployment of the DM in the ESLO corpus. 34
4.3 deployment of the DM in the TCOF corpus. 35
4.4 Example of mislabeling of the grammatical category by Unitex. 39

List of Tables

2.1 Description of the CamemBERT models and data used for training. . . 8
2.2 Description of the FlauBERT models in terms of layers , parameters and embedding dimension. 9

3.1 Statistics of *Le Monde* corpus. 25

4.1 Statistics on the CORPAIX corpus. 33
4.2 Statistics on the ESLO corpus. 34
4.3 Statistics on the TCOF corpus. 35
4.4 Evaluation of Unitex with its internal resources on the identification of marker *bon* in 100 samples from the CORPAIX data. 36
4.5 Evaluation of Unitex with external resources on the identification of marker *bon*in 100 samples from the CORPAIX data. 36
4.6 Distribution of the evaluated DM in the three corpora. 36
4.7 Statistics about the evaluated data. 36
4.8 Results obtained from the CORPAIX data. 36
4.9 Results obtained from the ESLO data. 37
4.10 Results obtained from the TCOF data. 37
4.11 Distribution of samples in train, dev and test data for DM *attention* and *bon*. 37
4.12 Results of the pre-trained model for DM *attention* and *bon*. 37
4.13 Statistics about the evaluated data in order to classify the DM *attention* in the three datasets. 38
4.14 Results obtained for the categorization of DM *attention*. 38

Chapter 1

Introduction

Discourse consists of collocated, ordered, and coherent clusters of sentences, instead of single and unrelated sentences. Discourse is coherent if among other things there are significant coherence relations between the utterances. For example, to justify the connection between utterances, some type of explanation could be required. One of the lexical phenomena that ensures coherence in discourse is the presence of a *discourse marker* (DM).

DM are linguistic expressions that have been proven to be effective for (a) segmenting discourse into meaningful units and (b) recognizing relationships between these units. Determining the meanings of discourse markers is frequently necessary for understanding the message (Petukhova and Bunt, 2009).

Some DM, called *connectors* (DMC), can express semantic relations between discourse segments. For example, *because* expresses cause/explanation, *but* expresses contrast, etc. Other DM, called *particles* (DMP), can express speakers' attitudes, for instance their emotional or belief state, and play a role in the management of interaction, etc. The best-known examples are emotional interjections like *oh* or *ah* , but there are many other attitudinal DM, like *bon, tu parles, quoi, hein, tu vois, écoute*, etc. in French, whose meaning is more difficult to describe.

The most difficult aspect of DM identification lies in the fact that the category of markers is fairly clearly a functional-pragmatic, and not a formal, morphosyntactic one (Lamiroy and Swiggers, 1991). DM can come from a variety of distributional classes, and they frequently have formally identical counterparts that are not used as markers but do contribute to propositional content (whereas markers don't), as shown in the following examples , in (1) *bon* acts as a NON DM (adjective) and in (2) acts as a DM.

(1) Nous avons passé/sommes restés un bon moment chez nos voisins.
We stayed with a good/long time at our neighbors.

(2) A – je vais te faire un super cadeau pour ta fête.
B – bon j'ai hâte de voir ça
A – I'll give you a great gift for your party.
B – Well, I cannot wait to see that

1.1 Main Question

Some discourse markers are routinely observed as having many functions, and called *polyfunctional* [1] DM. Moreover, the further along the grammaticalization line they have progressed, the more functions they appear capable to adopt.

When viewed from the standpoint of automatic discourse analysis, the feature of polyfunctionality is important. Indeed, the DM can allow to detect the structure and interpretation of a discourse at a local level (i.e., choose between multiple possible analyses), reducing the size of the analysis forests dramatically.

A resolution mechanism is therefore necessary in the perspective of automatic processing of discourse to identify whether those polyfunctional items act as DM or as part of the *descriptive* content[2] At the moment, there is no existing mechanism for this purpose, and the work we present here aims to fill this gap. In addition, the idea behind studying a spoken corpora is to explore possible functions of DM due to their frequency and the particular structure of this type of corpora.

1.2 Host Organization

The internship was completed at ATILF - Analyse et Traitement Informatique de la Langue Française, which is a linguistics and language studies research laboratory (UMR 7118, CNRS). ATILF is a collaborative research unit of the CNRS (French National Centre for Scientific Research) and the University of Lorraine in France. My supervisors are Dr. Mathilde DARGNAT, Assistant Professor at University of Lorraine and member of the ATILF lab, and Dr. Mathieu CONSTANT, Professor at University of Lorraine and member of the ATILF lab too. Jacques JAYEZ, Emeritus Professor at ENS de Lyon and member of the LORIA Sémagramme team, acted as a co-advisor.

ATILF has 135 members, 5 research teams on various subjects, and 4 support services, including a documentation center called Michel Dinet, at the time of writing this report. The ATILF's research is divided into five groups: Lexis, Historical linguistics for French and Romance languages, Discourse, Applied linguistics and sociolinguistics (Crapel), and Normalization of Resources, annotation and use.

The intership topic was chosen in the context of an ANR project submission about discourse marker collocations and (non-)compositionality. I did my internship on-site at the laboratory and via web conferencing during the covid time. In addition, I got the opportunity to participate in the Café TAL, which is a weekly event in which members of the lab can present their work or discuss questions in the NLP domain.

The coarse timeline of the internship is defined below:

- Literature survey and practice tools : from March to May.
- Implementation : May to August.
- Report writing : August.

[1] Polyfonctional DM's list existing in :https://github.com/Dahouabdelhalim/Discourse-marksers-and-Web-crawling/.The process of extracting thos DMs is based on finding those who are in intersection between Charlotte Roze's dictionary and DELA-fr and having another category in DELA-fr.

[2] The descriptive content corresponds to the truth-conditional part of the linguistic message. For instance, in a sentence like *Bon, j'ai raté mon bus* or *Well, I missed my bus*, the sentences *j'ai raté mon bus* or *I missed my bus* convey the descriptive content. Other terminologies exist: *main* content, *propositional* content or *at issue* content.

1.3 Structure of the Report

This document is organized as follows: In Chapter 2, I summarize the notions of discourse marker identification and annotation, as well as previous work in the field of discourse markers and polyfunctional discourse markers. In Chapter 3, I discuss the various techniques used to achieve the goal, the report's primary work is detailed in Section 3.1 , 3.2 and 3.3 and 3.4. The data in Chapter 4 will be presented in numerical and plot forms, together with the experiments that were used and the results for each experiment with discussion about the outcomes of each methods. Finally , In Chapter 5, we compare and contrast the methodologies used in this study, as well as some drawbacks, issues, and future research directions.

Chapter 2

State Of The Field

2.1 Discourse and discourse markers

As mentioned in the introduction, discourse markers (DM) help us to assemble discourse segments into more or less coherent chunks by highlighting various discourse relations and attitudes of the speaker/author. There is by now a vast descriptive and theoretical linguistic literature on DM (see Dostie (2004) and Brinton (2017) for some examples), addressing in particular the questions of their semantic properties, their role in the organization of texts and conversation and their evolution. It is generally acknowledged that some discourse relations such as cause, explanation, consequence, anteriority, posteriority, simultaneity, contrast, condition, etc., play an important role in the production and interpretation of meaningful texts. Techniques have been developed to identify and annotate such discourse relations in explicit formal frameworks (Stede, 2012).

However, the situation is not the same when one includes lexical elements which denote more complex semantic constructs, typically emotional or intellectual attitudes, signals of speech monitoring, etc. These elements, called *particles*, presented in Dargnat (2021), are mostly found in spoken corpora, whose structure is often less apparent than with written corpora. Their semantic identity is also often difficult to describe. For instance, what is the 'meaning' of the particle *bon* in French?

Particles can be considered as adverbials adjuncts. As noted in the introduction, some of them can also be categorized as nouns, adjectives or verbs in other contexts. Although human experts have in general no problem in distinguishing the adverbial and non-adverbial function of a word, this is not the case for machines, which makes the most basic task of collecting particles in a corpus a bit hazardous.

2.2 Previous Related Work

The problem of discriminating functions for particles is not new (Zufferey and Popescu-Belis, 2004) but has remained marginal, possibly because it is much less crucial for written language. Actually, in recent literature, the discourse markers which are used as pivots to predict text continuations (Wu et al., 2020) or learn discourse relations (Nie, Bennett, and Goodman, 2019) are 'standard' DM like *because* or *but*, and not particles. A more liberal inventory is used by Sileo et al. (2020) but is not intended to address the ambiguity problem. To our best knowledge, there is no systematic attempt to tackle this issue in recent work.

2.3 Resources used

2.3.1 Unitex platform

Unitex/GramLab [1], abbreviated here as Unitex, is a multi-platform framework that consists of a series of programs designed to analyze natural language texts with the help of linguistic resources. These resources are Unitex's most powerful aspect, and it would be much less interesting without them.

Electronic dictionaries, grammars, and lexicon-grammar tables make up the linguistic resources, which were originally created for French at the Laboratoire d'Automatique Documentaire et Linguistique (LADL), but similar resources have been developed for other languages as part of the RELEX laboratory network.

Unitex is made up of a Java-based graphical user interface and C/C++-based external programs. This combination of programming languages results in a fast and portable framework that runs on a variety of operating systems. In addition , it provides portability, modularity, and the ability to work with languages that use special writing systems to the user. Unitex offers the following strong points :

Open Source : The Lesser General Public License (LGPL) allows Unitex to be freely distributed . This means that, as long as the LGPL license is followed, anyone can freely redistribute Unitex.It also means you have access to every Unitex programs' source code. The LGPL license is more permissive than the GPL license in that it permits you to utilize Unitex/GramLab's code in non-free software.

Cross-platform : The Visual IDE is implemented in Java, whereas the Unitex Core NLP Engine is written in C++. This enables you to create Unitex-based programs on any machine that supports Java 1.7, compile them with any standard C++ compiler, and run them on your preferred platform: Windows, Linux, MacOS, and a variety of additional operating systems are available.

Multilingual : Unitex complies with the Unicode 3.0 standard, which enables users to manipulate practically all characters from any language, including Asian languages. The Unitex programs were created to function with all types of writing rules. Working with Asian languages is straightforward, despite their unique spacing conventions.

Lexicon-based : Unitex interacts with electronic dictionaries developed by the RELEX network members, an international network of Computational Linguistics laboratories founded by Maurice Gross and his LADL team. For many of the LGPLLR-licensed resources offered with Unitex/GramLab, members of the RELEX network have constructed or are working on comprehensive dictionaries. For the French language, Unitex contains a demonstration corpus, *Le tour du monde en quatre-vingt jours* by Jules Verne. For the dictionaries, it has 683,824 simple words (102,073 distinct lemmas) , 108,436 compound words (83,604 distinct lemmas), given name dictionaries (24,000 entries) , profession dictionary (4,200 entries) and 2,700 Quebec simple words.

Grammar-based : Local grammars are a useful way to express syntactic and semantic rules. It uses finite state automata and electronic dictionaries to perform textual data analysis automatically. The benefit is to quickly develop, test, debug, maintain, and apply local grammars on a text using Unitex visual IDE.

Text preprocessing : Unitex offers preprocessing operations on the input text in order to remove ambiguity , defining the test units , and providing grammatical , semantic and flexionnels information. Those operations contains :

[1] https://unitexgramlab.org/

- Normalization of separators : Separators taken into consideration in this case are the space, the tab and the newline characters because their presence may have an effect on the process of splitting the text into sentences.
- Splitting into sentences : Splitting texts into sentences is a vital preprocessing phase because it aids in defining the units for linguistic processing and text automaton development. Since detecting sentence boundaries is not a simple task, some grammars are used in this process.
- Normalization of non-ambiguous forms: The application of this step is due to the existence of certain forms in texts that need to be replaced by a corrected (often expanded) form (for example, the English sequence "I'm" is equivalent to "I am"). One may want to replace these forms according to one's own needs.
- Splitting a text into tokens : Unitex splits texts in a language-dependent way because there are some languages, in particular Asian languages, that use separators that are different from the ones used in Western languages. The tokenization applied by Unitex is case-sensitive ('A' and 'a' are two distinct tokens), and finally the text is represented by a sequence of numbers (integers) that describe each token in the text just once. This process creates a few files that are saved in the text directory.
- Applying dictionaries : Unitex gives the possibility to apply some dictionaries with the goal of building other 'sub dictionaries' consisting only of forms that are present in the text.

Based on the preceding, we are particularly interested in employing the lexicon-based feature in this study in order to learn more about the DMs and their context. Furthermore, the grammar-based functionality will make it easier for us to identify DMs by stating some rules using Unitex's simple interface. (cf. section 3.1))

2.3.2 CamemBERT

CamemBERT (Martin et al., 2019) is a pre-trained language model and a French version of the Bi-directional Encoders for Transformers (BERT). The idea behind the invention of this pre-trained model was to remedy the fact that most available models have either been trained on English data or on the concatenation of data in multiple languages, which means that the use of such models in all languages except English is very limited. CamemBERT is intended to foster research and downstream applications for French NLP. This pretrained model improves the state of the art in multiple downstream tasks, namely part-of-speech tagging, dependency parsing, named-entity recognition, and natural language inference compared to multilingual models.

The benefits of pre-trained language models is that they are being trained on large datasets and for a long time (significant number of epochs). The authors of CamemBERT used the French part of the OSCAR corpus (Ortiz Suárez et al., 2019), which is a set of monolingual corpora extracted from Common Crawl snapshots and pre-filtered and pre-classified, another Common Crawl extract named CCN (Wenzek et al., 2019) and a recent snapshot of the French Wikipedia. For each corpus, they created a version of CamemBERT in order to compare equitably the impact of the pre-training data and to also study the effects of the size of the training data on model performance. The following table summarizes the characteristics of each version from the CamemBERT model.

Model	params	Arch.	Training data
CamemBERT-base	110M	Base	OSCAR (138 GB of text)
CamemBERT-large	335M	Large	CCNet (135 GB of text)
CamemBERT-base-ccnet	110M	Base	CCNet (135 GB of text)
CamemBERT-base-wikipedia-4gb	110M	Base	Wikipedia (4 GB of text)
CamemBERT-base-oscar-4gb	110M	Base	Subsample of OSCAR (4 GB of text)
CamemBERT-base-ccnet-4gb	110M	Base	Subsample of CCNet (4 GB of text)

TABLE 2.1: Description of the CamemBERT models and data used for training.

We can see two different architectures for CamemBERT: the original CamemBERT model, which was trained with 12 layers, 768 hidden dimensions, and 12 heads of attention, resulting in 110m parameters. The large model is broad CamemBERT, which was trained with 24 layers, 1024 hidden dimensions, and 16 attention heads, for a total of 340m parameters.

. Finally, we conclude with some usage of this pretrained model specially for the French language.

- To complete a sentence / masked LM (MLM).
- Write a complete article / Next Sentence Prediction (NSP).
- Compare the meaning of two sentences.
- Determine the subject and complement(s) of the verb,.
- Named Entity recognition.

2.3.3 FlauBERT

FlauBERT (Le et al., 2019) appeared a few weeks after CamemBERT. FlauBERT has also released a new Francophone NLP evaluation reference (FLUE), which can be used to compare the performance of various models. They employ a design similar to that of CamemBERT to train FlauBERT (and therefore BERT). We can define FlauBERT as a collection of models trained on a big, varied French corpus with the use of the new CNRS Jean Zay Supercomputer in order to run models of various complexity. The test of the language models was done on FLUE (French Language Understanding Evaluation) which is an evaluation setup for French NLP systems that is analogous to the prominent GLUE benchmark. The purpose is to make future experiments more repeatable, as well as to exchange models and development on the French language.
FlauBERT was built using data from 24 sub-corpora obtained from various sources, covering a wide range of topics and writing styles, from formal to well-written texts. The data was obtained from three main sources: monolingual French data from WMT19 shared tasks, French text corpora from the OPUS collection, and datasets from Wikimedia projects. They utilized their own tool to extract the text from the other sub-corpora or downloaded it directly from their websites. As mentioned in their paper , before preprocessing, the uncompressed text is 270 GB in size.

Based on the results presented in the original paper, FlauBERT is competitive with CamemBERT, which is being trained on almost twice as fewer text data compared to FlauBERT. We conclude the description with some NLP tasks managed by FlauBERT models :

Model name	Nb. layers	Attention Heads	emb. dim	Total Parameters
FlauBERT-small-cased	6	8	512	54 M
FlauBERT-base-uncased	12	12	768	137 M
FlauBERT-base-cased	12	12	768	138 M
FlauBERT-large-cased	24	16	1024	373 M

TABLE 2.2: Description of the FlauBERT models in terms of layers , parameters and embedding dimension.

– Text classification.

– Paraphrasing.

– Natural language inference (NLI).

– Parsing .

– Word sense disambiguation.

Chapter 3

Methodology

As mentioned in section 1.1, the goal of this study is to identify some polyfunctional DMs in French spoken corpora. In the literature, the identification process of those polyfunctional DMs is not dependent just on the DM expression. According to Lamiroy and Swiggers (1991, p. 123) " the category of markers is fairly clearly a functional-pragmatic, and not a formal, morphosyntactic one", which means that we need to take the context or the environment of those DMs in consideration to identify their function.
Given that, the most suitable approaches for reaching our research goal are the rule-based approach and the machine learning approach.

The first approach is the oldest. It is a tried and tested approach which has proven its efficiency through its results, especially for studying linguistic phenomena. In addition, it is relatively easy to build and apply some pattern-matching or parsing that helps to check the context of DM and delivers good performance when used on specific cases.
The second approach is based on neural network architectures which have been successfully applied to a broad set of problems in various areas of natural language processing. With the recent success of word embeddings (low dimensional, distributed representations), neural-based models have achieved superior results on various language-related tasks, in particular when this approach can construct word vectors that embed syntactical and semantic information dependent on the word context.

As shown in Figure 3.1, each strategy described in this part will borrow some features from one of the approaches mentioned above. As can be seen, each approach has two techniques, each of which is based on a distinct hypothesis. It is thought that investigating this topic from multiple perspectives yields more reliable, convincing results, as well as better research perspectives.

The first rule-based method will use the Unitex platform's local grammars to express certain syntactic rules for each DM depending on the left and right positive context of the DM. The second will be based on the DM's left and right negative contexts by generating syntactic and lexical patterns derived from the *Le Monde* corpus and applying them in matching mode to the spoken corpora.

In contrast, the first machine learning technique focuses on learning automatically from a training corpus that contains data with their POS labeling, in order to determine the correct function of potential DM (real DM or other categories, such as noun, adjective, etc.). The second tries to determine the function of DM using the KNN method. It takes as input the target DM's word embedding vector and classifies it using the similarity distance.

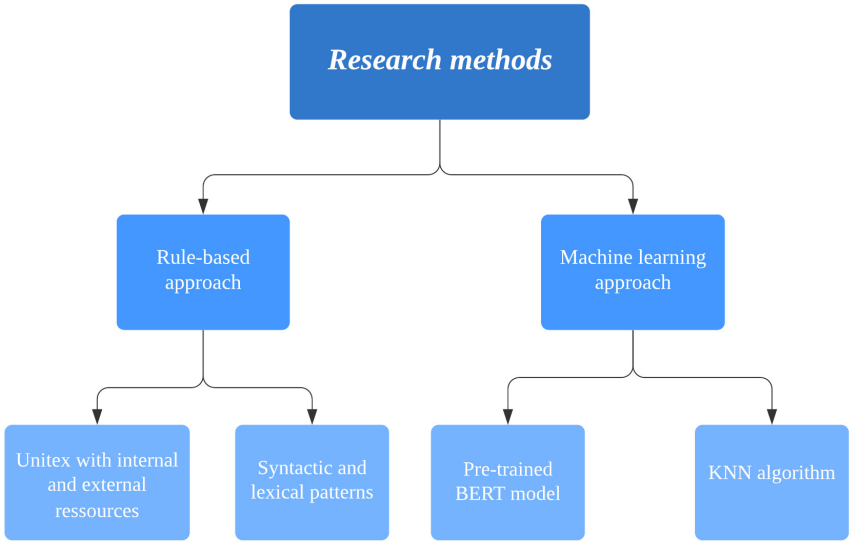

FIGURE 3.1: illustration of the methods used in this study with their variants.

3.1 Unitex with internal and external ressources

In this section, we describe the processing pipeline of Unitex, its features and resources, how to use the theme and a comparison scenario in order to avoid the drawbacks of Unitex and to increase the performance of the system.
The first scenario, illustrated in figure 3.2 , consists in using the internal resources of Unitex without any external intervention. The second scenario illustrated in figure 3.12 consists in using external resources, bypassing the resources offered by Unitex such as its dictionaries and linearized tagger, in order to estimate the effect of Unitex resources and explore its drawbacks.

3.1.1 Unitex with internal resources

This method focuses on identifying the function of potential DM in a spoken corpus using just Unitex's resources (dictionary and linearized tagger) and local grammars (graphs) that describe syntactic rules sensitive to the left and right positive context for each DM as described in 3.2.

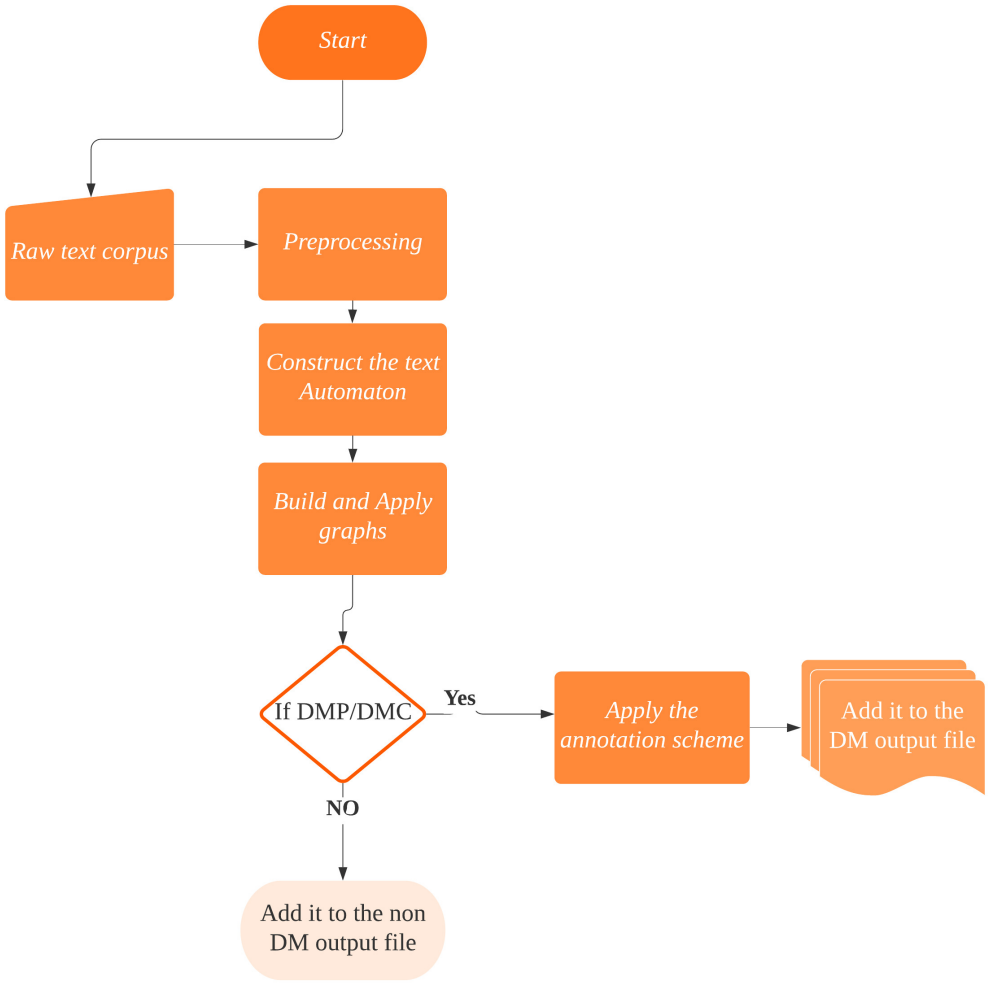

FIGURE 3.2: Unitex with internal resources pipeline .

Read and preprocess the spoken corpus

Unitex works with a variety of document kinds and supports a variety of text formats. In our case, we are dealing with.txt files in UTF-8 format. Unitex reads the corpus in the same way as other applications do.
To start working with the selected text , Unitex offers to preprocess it. The preprocessing operations are the same as the steps mentioned in section 2.3.1.
After finishing applying the preprocessing operations , Unitex opens a new window with the sorted lists of simple, compound, and unknown words found after the dictionary look-up is completed as shown in the figure 3.3.

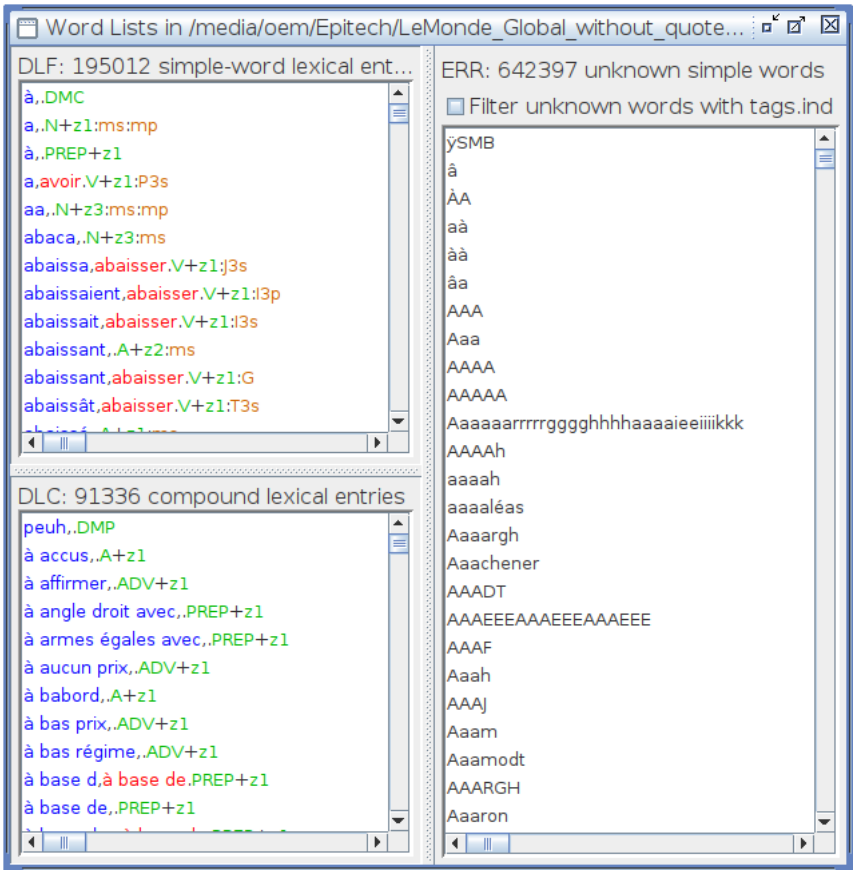

FIGURE 3.3: Result of applying dictionaries to a text.

The electronic dictionaries distributed with Unitex use the DELA syntax (Dictionnaires Electroniques du LADL, LADL electronic dictionaries). They describe the syntax of simple and compound lexical entries of a language by providing grammatical, semantic and inflectional information. We can find for one and the same word different entries with different grammatical, semantic or inflectional formats. After applying this step, Unitex will create three files that are saved in the text directory (dlf for simple words, dlc for compound words and err for unknown words).

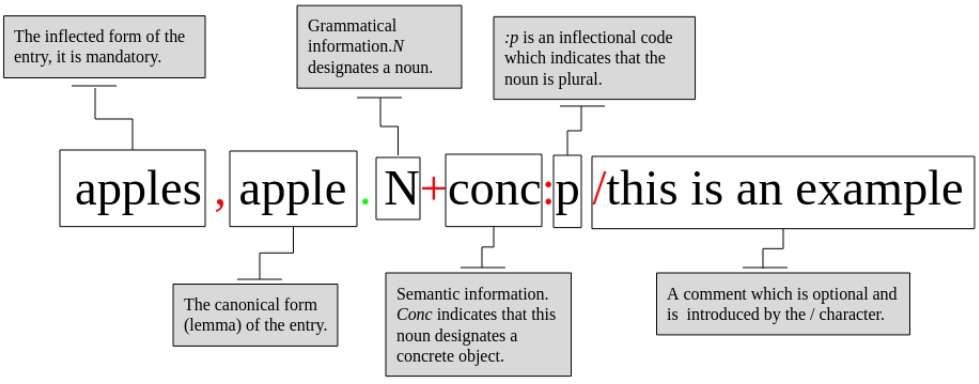

FIGURE 3.4: DELAF syntax example.

Construct the text automaton

Unitex offers the possibility of constructing a text automaton. In fact, the meaning of "text automaton" is that Unitex provides an automaton for each sentence of the text. It is well-known that natural languages contain much lexical ambiguity. The text automaton is an effective and visual way of representing such ambiguities. Each sentence of a text is represented by an automaton whose paths represent all possible interpretations.
The text automaton explicates all possible lexical interpretations of the words as shown in figure 3.5. These different interpretations are the different entries presented in the dictionary of the text.[1]

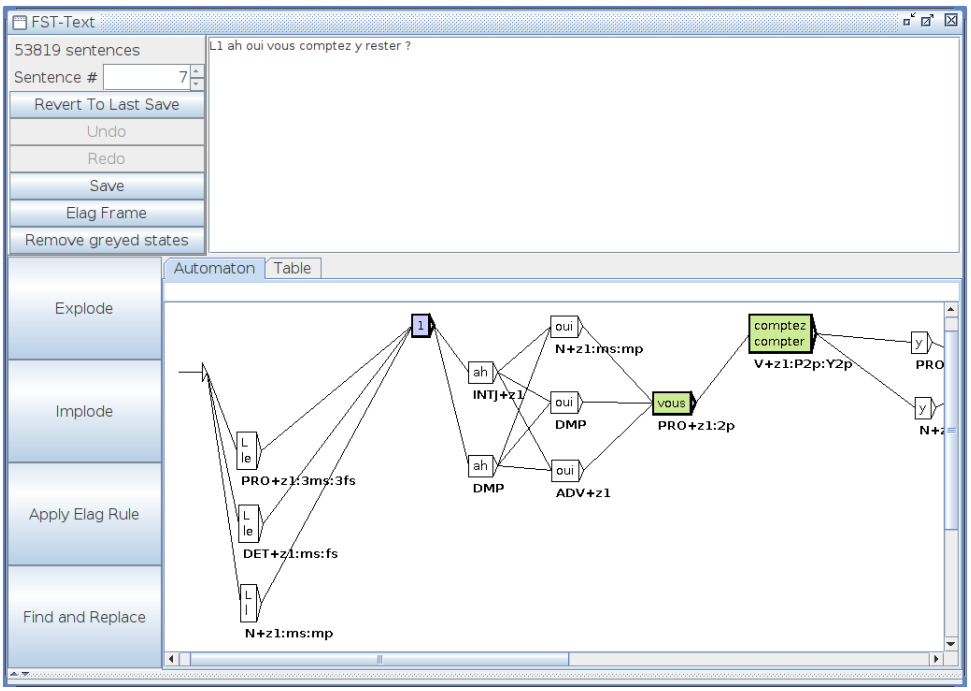

FIGURE 3.5: Sentence automaton example.

As shown in Figure 3.5 , the text automaton contains many paths of tags because of the lexical ambiguity in the dictionaries. Working with those interpretation paths will affect the results of searching and the identification of the right category of the token. For that the use of a linearization process is crucial.
The linearization process consists in selecting a single path, a sequence of tags with one tag per token, and removing the others. The output of the process is a text automaton with a single path . The selection of a path depends on its score. The path with the best score is chosen and the others are removed. The score of a path is calculated using a statistical model trained on an annotated corpus. This model uses tagger data files generated by the TrainingTagger program integrated with the unitex platform.
For instance, we can see on 3.5 , the original text automaton of the French sentence *L1 ah oui vous comptez y rester ?*. The corresponding text automaton after linearization is shown on Figure 3.6.

[1]The text automaton construction should be applied after the preprocessing step because, if no dictionaries are applied, the resulting text automaton will consist of only one path made up of unknown words per sentence.

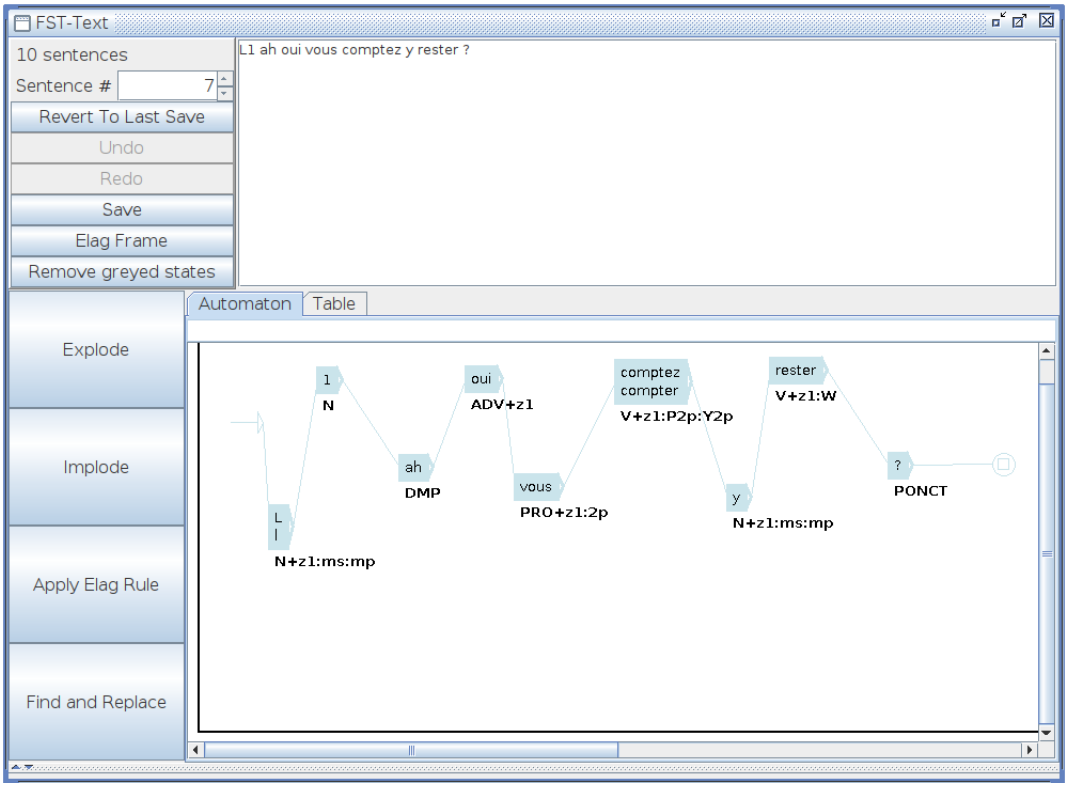

FIGURE 3.6: Text automaton linearized.

Build and apply graphs

To represent our linguistic phenomena, we will use local grammars, in the form of visual graphs, for which there are powerful tools provided by Unitex. Those graphs have some properties such as producing an output , using other graphs as sub-graphs and managing the priority between graphs. The most important feature is the possible management of the left and right contexts. The arrow symbol refers to the initial state of the graph, the round symbol with a square to the final state of the graph. The grammar only recognizes expressions that are described along the paths between initial and final states.
Based on that , for each studied marker we will define the positive left and right contexts between the initial state and the final state as illustrated in figure 3.7.
The positive context is the context where the polyfunctional DM acts as a DM and not as a descriptive content part such as noun, adjective or pronoun in the case of *quoi*.

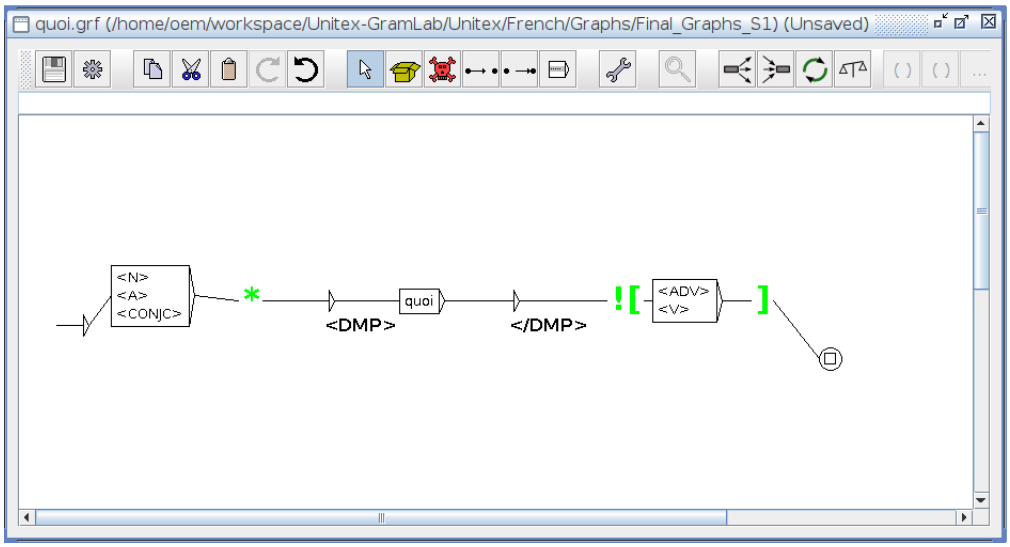

FIGURE 3.7: Graph of the marker *quoi*.

The left box represents the left context that must appear before the marker *quoi*, which contains three grammar categories which are respectively Noun, Adjective and Coordinating Conjunction.
The right box surrounded by the green square bracket represents the right context which contains two grammar categories which are respectively Adverb and Verb. The exclamation mark means the negation of the box.
The <DMP> and </DMP> expressions are the annotation scheme that will be applied by the graph in the matching case.
To sum up, the graph matches the word *quoi* preceded by Noun , Adjective or Coordinating Conjunction and followed by any grammar category except Adverb or Verb. The graphs can be found in our Github [2] repository.

[2] https://github.com/Dahouabdelhalim/Discourse-marksers-and-Web-crawling

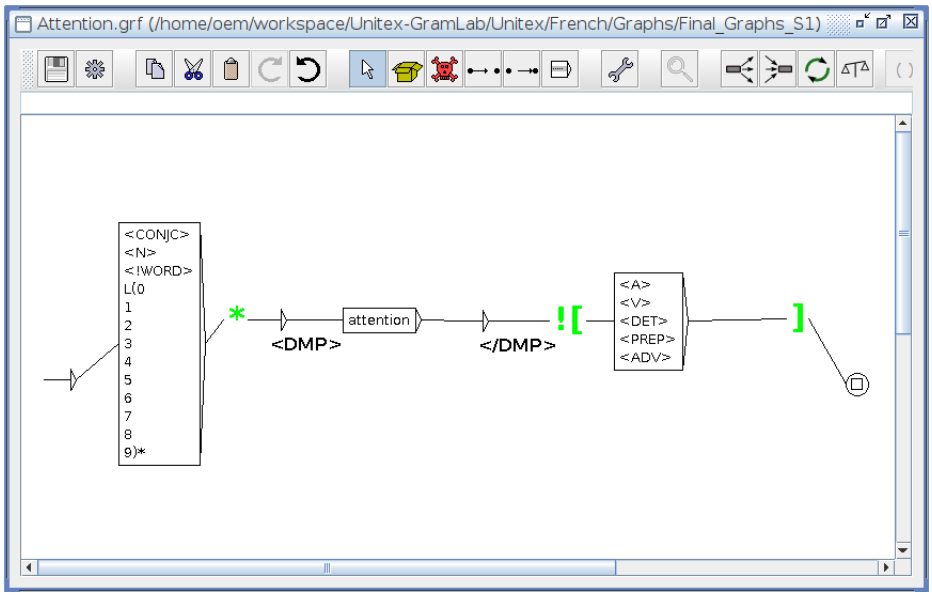

FIGURE 3.8: Graph of the marker *attention*.

To arrive at those final graphs 3.8, 3.9, 3.10,[3] we ran a series of trials to cover as many environments as possible for each DM. Analyzing each DM's behavior in the spoken corpus, as well as the various contexts that can arise in it, allows us to better define the positive context.

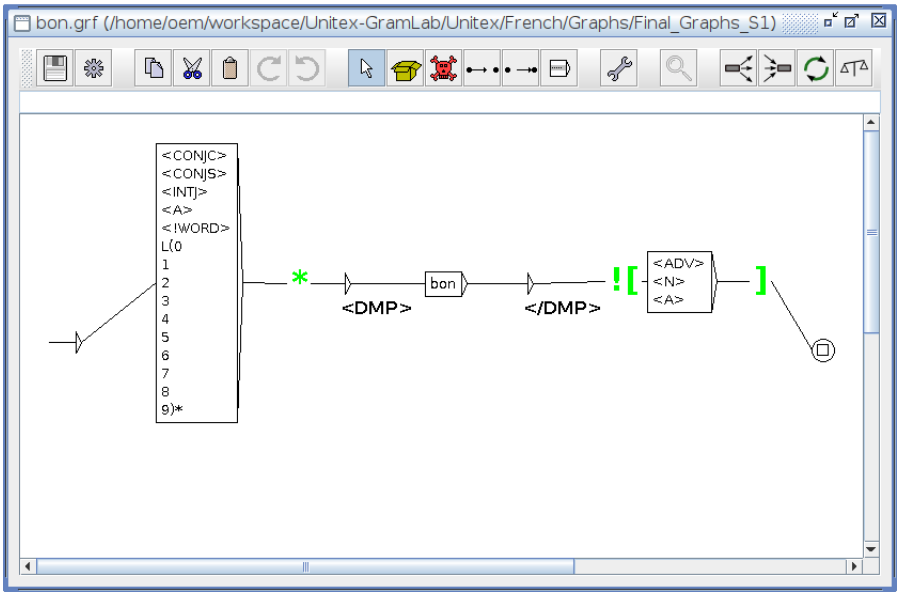

FIGURE 3.9: Graph of the marker *bon*.

[3]Codes: N = Noun, A = Adjective, V = Verb, ADV = Adverb, CONJC = Coordinating Conjunction, CONJS = Subordinating Conjunction, DET = Determiner, INTJ = Interjection, !WORD = non-(lexical word), L0, L1, etc. = speaker's number at the beginning of a speaking turn.

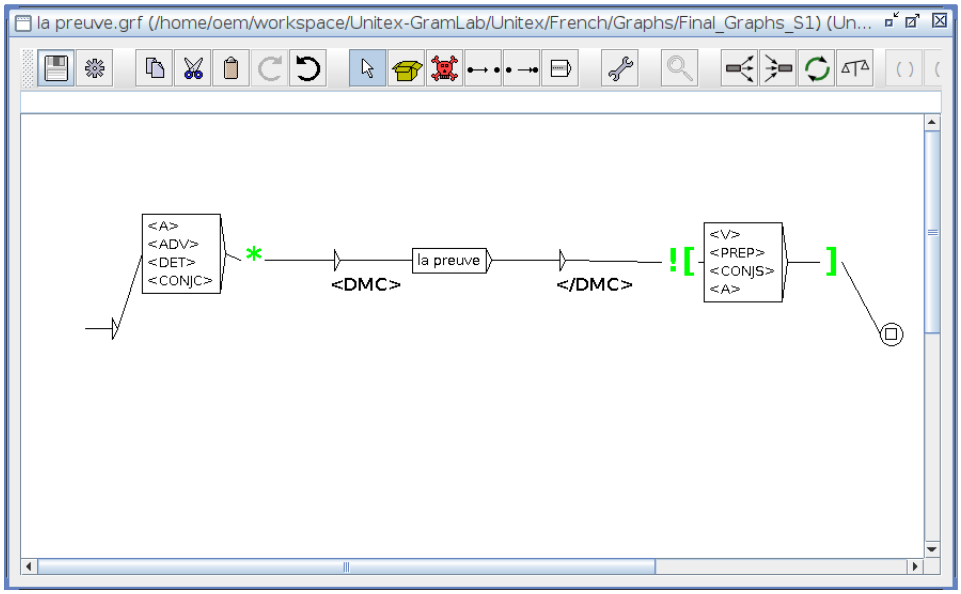

FIGURE 3.10: Graph of the marker *la preuve*.

After compiling the graph, we can search a text for all the sequences that match the pattern defined in the graph. Using various options of the "Locate Pattern..." command in the "Text" menu, We asked Unitex (1) to consider all the matching sequences, (2) to base its exploration on the text automaton (high precision), (3) to keep the <DMP> tags in the result, and (3) to store the matching occurrences with a left and right context of 100 characters. Unitex can then produce an appropriate concordance table in an .html file. Figure 3.11 shows the results for *quoi*.

FIGURE 3.11: Concordance obtained with the grammar of Figure 3.7

3.1.2 Unitex with external resources

The second scenario is focused on the same hypothesis as the one mentioned in section 3.1.1. The difference here is that we avoid using Unitex's resources (dictionaries and linearized tagger) and use an external resource, the camemBERT parser, as illustrated in figure 3.12. The parser helps us to determine the syntactic nature of each token in the text. We applied this scenario in order to : (1) evaluate the effect of Unitex's resources; (2) try to have a better identification precision in determining the function of the studied potential DM.

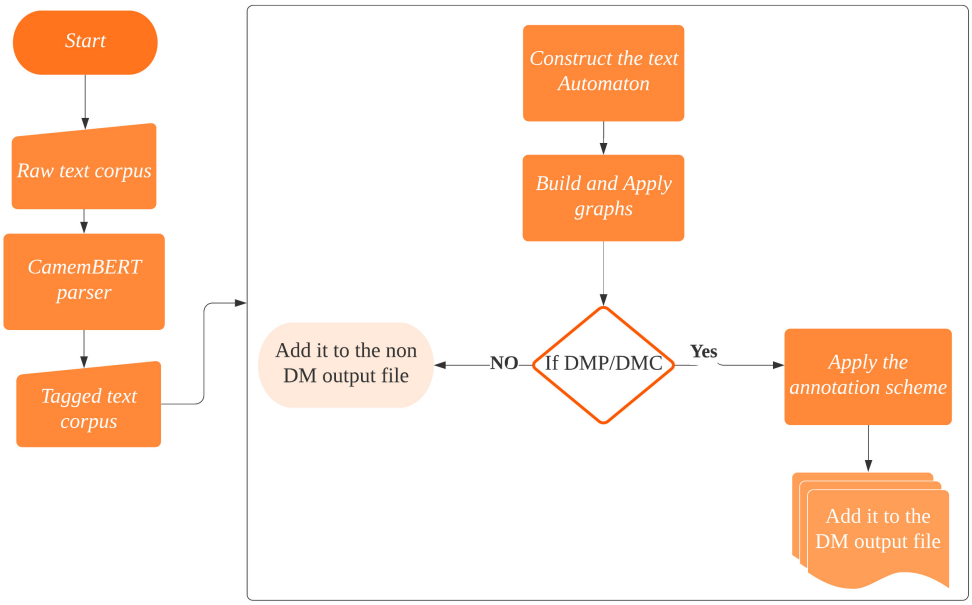

FIGURE 3.12: Unitex with external resources pipeline .

Unitex reading tagged corpus

A tagged text is a text containing words with extracted information such as grammatical tags enclosed in braces (Figure 3.13). Such tags can be used to avoid ambiguities. However, the presence of these tags can alter the application of preprocessing graphs. To avoid complications, we used the feature of using the "Open Tagged Text..." command in the "Text" menu in Unitex. With it, we can open a tagged text and skip the application of preprocessing graphs.

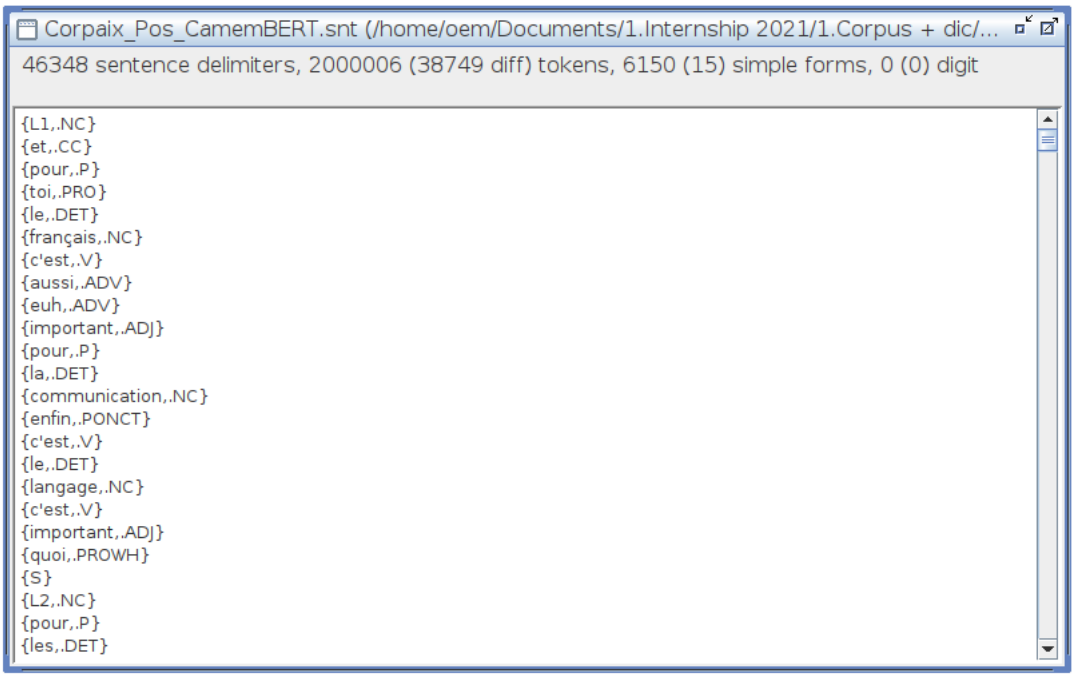

FIGURE 3.13: Tagged text with CamemBERT.

Construct the text automaton

This step is identical to the step presented in section 3.1.1.

Build and apply graphs

The difference between this step and the step presented in section 3.1.1 just concerns the tags used to specify the category of the left and right contexts. The CamemBERT parser used different tags to identify the grammatical category of the token such as CC for coordinating conjunction, ET for foreign word , NC for common noun and VINF for infinitive verb. The graphs remain the same, but we apply a conversion between the tags used in Unitex dictionaries and the tags used by the CamemBERT parser. The graphs mentioned in section 3.1.1 are converted into graphs that use the tags of the CamemBERT parser.

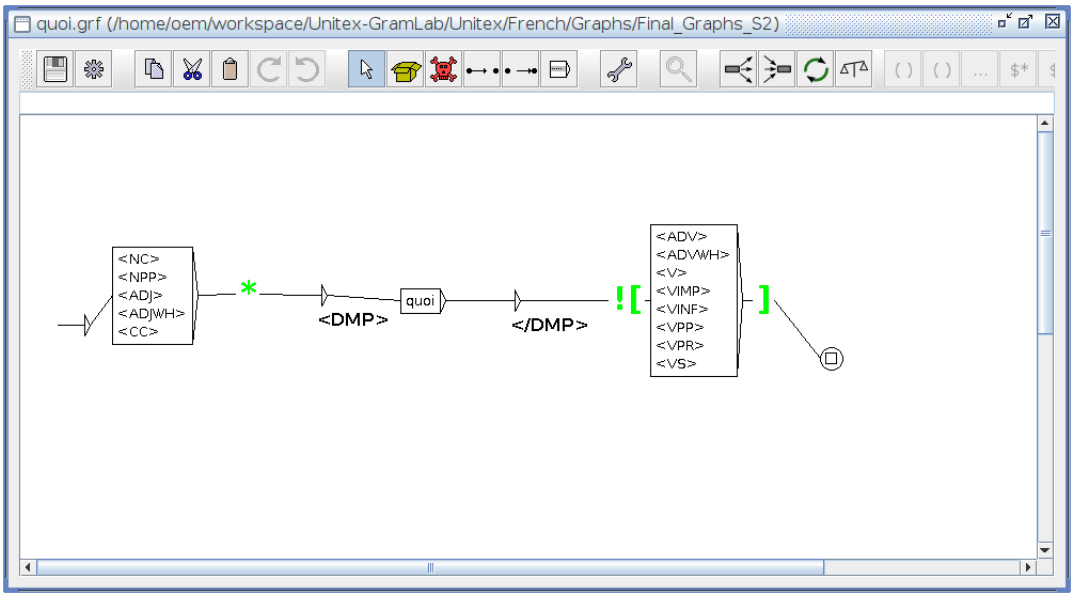

FIGURE 3.14: Graph of the marker *Quoi* with CamemBERT tags.

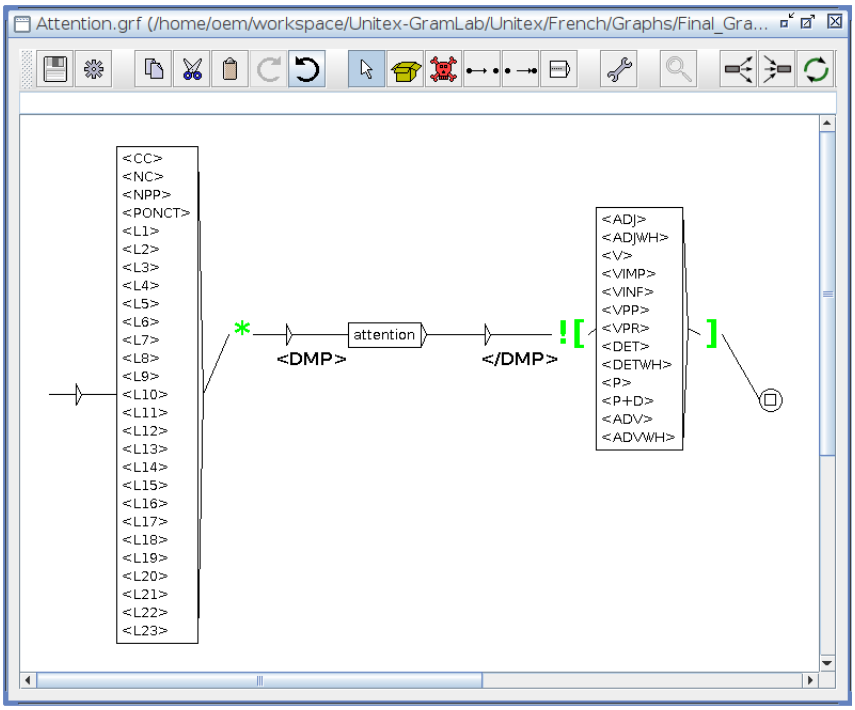

FIGURE 3.15: Graph of the marker *Attention* with CamemBERT tags.

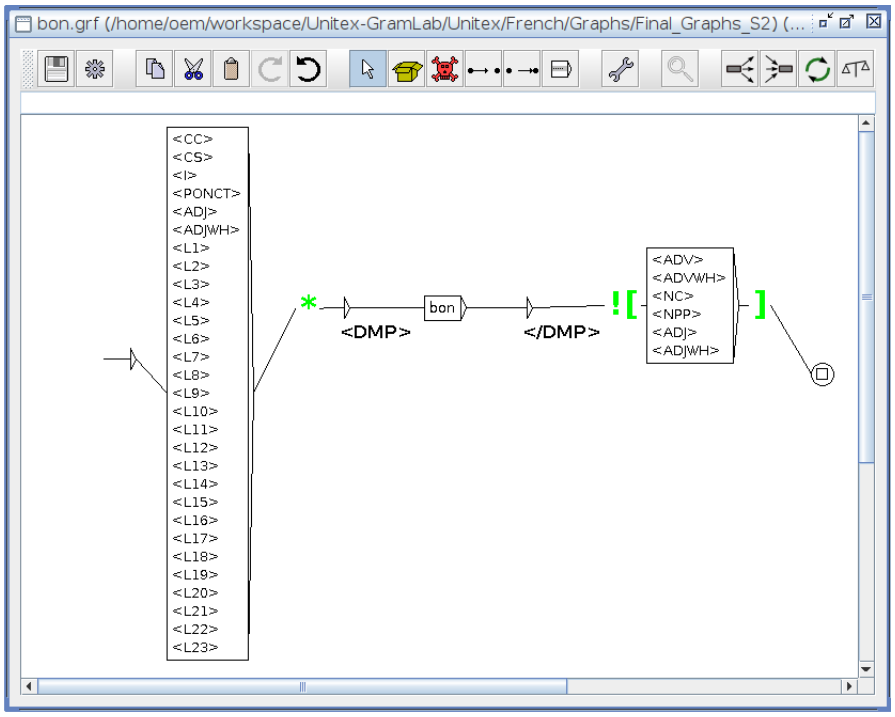

FIGURE 3.16: Graph of the marker *bon* with CamemBERT tags.

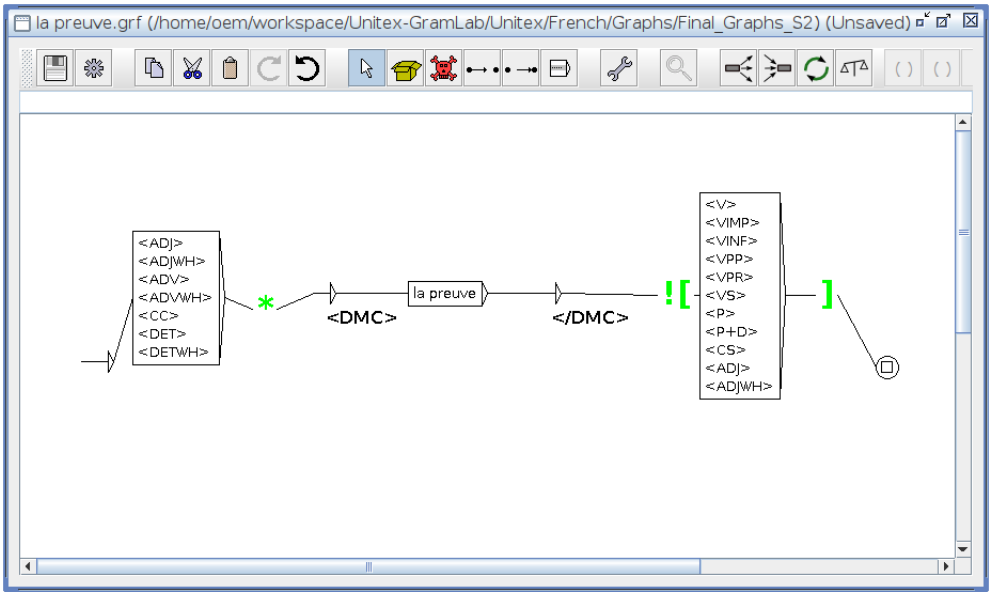

FIGURE 3.17: Graph of the marker *La preuve* with CamemBERT tags.

The process of applying the graphs and extracting the concordance is still the same as the one presented in section 3.1.1.

3.2 Syntactic and lexical patterns

In this section, we present the second rule-based method which has less expert intervention compared to the previous rule-based method.

The idea behind this method is to use the syntactic structure of the left and right context of the token in a 'negative' way, to reach a better precision in identification. Using negative contexts with lexical forms impose a huge knowledge base to support different environments. Since the items under study are to be found in spoken language, our negative comparison point should be a big set of sentences in written language. If a potential DM occurs in such a set there is a great chance that it is not a real DM. For that , we used the *Le Monde* corpus in order to construct a knowledge base (patterns).
The structure of this section will respect the steps of the processing pipeline presented in the figure 3.18 with details about reasons and resources used for each step. The process consists in building a list of patterns that holds the negative contexts of the potential DM item in the *Le Monde* corpus and applying this list to the spoken corpus. The matching between the pattern and the DM item in the spoken corpus means that the DM item acts as a NON DM, otherwise it's a DM.

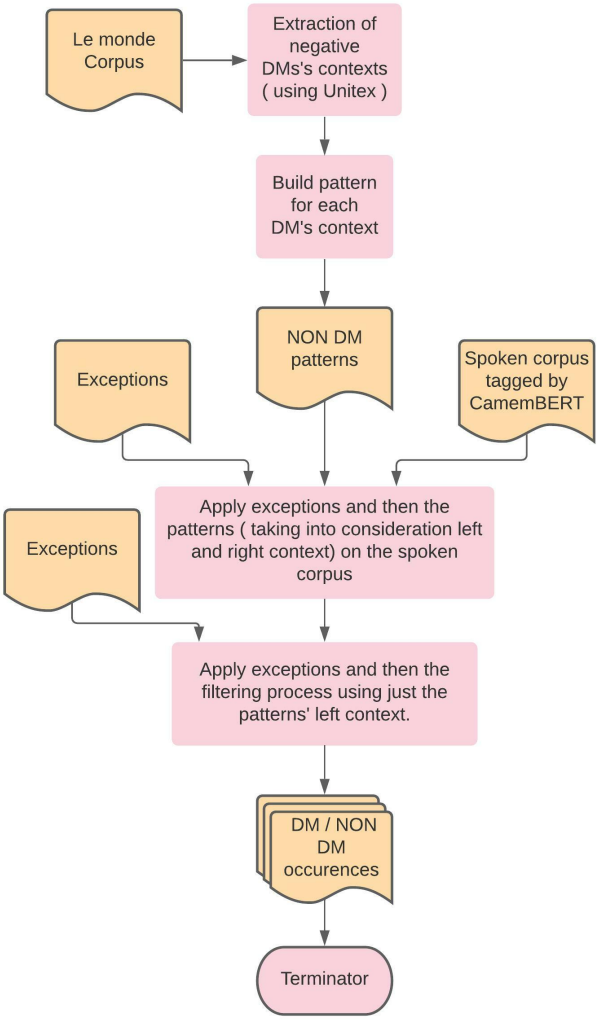

FIGURE 3.18: Syntactic and lexical patterns pipeline process.

3.2.1 *Le Monde* corpus

This corpus represents a secondary data that contains journal articles taken from the French newspaper *Le Monde* between 1987 and 1998. This newspaper was founded on 19 December 1944, and published continuously since its first edition. The papers correspond to thousands of articles and contain millions of words. According to Wikipedia, *Le Monde* was traditionally focused on offering analysis and opinion, as opposed to being a newspaper of record.
The motivation behind using this corpus is: (1) Availability and access; (2) large number of words and segments; (3) compliance with the standard French written language, a fact which can help us to find various negative contexts for the studied DM.

Tokens	**Sentences**	**Simple forms**	Compound forms	Size
894929	15242570	195012	91633	1,7 GB

TABLE 3.1: Statistics of *Le Monde* corpus.

3.2.2 DM negative context extraction

The goal of this step is to extract all the environments of the 'DM'[4] that exist in the corpus by using the Unitex platform. The procedure is the same mechanism as the one presented in section 3.1.1 with the following difference: the built graphs have no exceptions or limitations about the left and right contexts, which means that we search just for the occurrences of the DM under study in the *Le monde* corpus.
Unitex will read the corpus and directly start the preprocessing in order to segment , tokenize , apply the linguistic resources and generate the text automaton to get a high precision in the search phase. Then , we define the graphs for each DM candidate. Finally , we apply the graphs sequentially in order to generate the HTML files that contain the environments of each DM.

3.2.3 Patterns extractions using CamemBERT

The generated HTML files from the previous step will undergo a cleaning phase in order to extract the text content and remove the HTML tags to be ready to be used by the CamemBERT parser. After that , CamemBERT will generate a tagged text that assigns a syntactic category to each token. Finally, we organize each DM into a pattern that includes three words with their syntactic categories from the left and right contexts as shown in figure 3.19.

B	C	D	E	F	G	H	I
CL3	**CL2**	**CL1**	**WORD**	**CR1**	**CR2**	**CR3**	**Example**
peut , V	pas , ADV	faire , VINF	attention	tous , ADJ	ses , DET	enfants , NC	peut pas faire attention tous ses enfants
On , CLS-SU	doit , V	faire , VINF	attention	une , DET	dramatisation	de , P	On doit faire attention une dramatisation de
bon , ADJ	chat , NC	, , PONCT	bon	rat , NC	: , PONCT	plus , ADV	bon chat , bon rat : plus
ma , DET	tante , NC	a , V	bon	coeur , NC	, , PONCT	il , CLS-SUJ	ma tante a bon coeur , il
commencent	à , P	administrer ,	la preuve	que , CS	l' , DET	on , CLS	commencent à administrer la preuve que l' on
a , V	affirmé , VPP	avoir , VINF	la preuve	que , CS	cette , DET	opération , NC	a affirmé avoir la preuve que cette opération
tout , ADV	haut , ADJ	en , P	quoi	elle , CLS-SUJ	consiste , V	. , PONCT	tout haut en quoi elle consiste .
seule , ADJ	à , P	dire , VINF	quoi	que , CS	ce , PRO	soit , VS	seule à dire quoi que ce soit

FIGURE 3.19: The pattern extraction results that used to classify the DM and NON DM.

[4]Let us recall that, given that *Le Monde* is written language, most forms of *bon* or other markers are in fact not the DM but their functional variants, like the adjective *bon*.

3.2.4 Apply the patterns

Firstly, the CamemBERT parser will run over the spoken corpus in order to determine the syntactic category of each token depending on its context and, after that, we arrange the output as the pattern's structure.
Secondly, an exception list for 'attention' and 'bon ' is created which contains some multi words such as "alors attention", "mais bon", "enfin bon", "ah bon" and others. The goal here is to prevent spoken corpus occurrences from being classified incorrectly. Some forms, such as "alors attention", "mais bon" show that the DM operates as a DM and not as part of the descriptive content of the text. If we do not apply this exception process, the precision and recall will be affected negatively.
Finally, the classification procedure will begin by determining whether an item from the exception list exists in the spoken corpus occurrences; if so, we transfer the occurrences straight to the output DM file; otherwise, we use the patterns to determine whether the occurrence is a DM or a NON DM in matching mode and add it to its corresponding output file.

3.2.5 Filter process

The motivation behind the filter procedure arose from the expert's analysis of the preceding step's outcomes. The experts discovered that the patterns in use categorize correctly the NON DM occurrences, but that the potential DM occurrences gave rise a number of false positives (NON DM classified as DM).

Based on this, a filter procedure was applied on the DM occurrences in order to filter out the NON DM and store them in the appropriate file.
Firstly the process handled the exception as in the previous step and for the same reason. Secondly , when applying the patterns applying , the process starts with the same patterns but is limited to the left context for the nominal and pronominal DMs (*attention*, *la preuve* and *quoi*) and to the two left and right tokens for *bon*.
Summarizing , if an item from the exception list occurs in the spoken corpus occurrences, the process will keep the occurrence of this item in the DM file; otherwise, the patterns will be used to filter the occurrences (keep the DM occurrences and transfer the NON DM to the NON DM file).

3.3 Fine-tuning Pretrained Bert model for PoS Tagging task

The ability to take a model that has been pre-trained on massive datasets by companies like Google and OpenAI and apply it to more specific cases is a big reason for the success of transformer models. This is sometimes all we need , we just take the model and roll with it.
However, there are occasions when we need to fine-tune the model. Then, we will have to focus on our unique concrete case a little bit. Because each transformer model is different, and fine-tuning for various use cases is also different, we will concentrate on fine-tuning the $FlauBERT_{large}$.
The idea presented in this technique is to train a pre-trained model on a POS task, adding a new category named DM in order to determine whether the studied potential DM acts as a DM. The training data was constructed manually and annotated with grammatical tags by the spoken camemBERT parser. After that , we added the new tag ' DM ' to the items that acted as DM as shown in (3) for the *attention* DM; otherwise we keep the ordinary tag as shown in (4) for *attention*.

(3) L2 oui mais attention
NC ADV C DM

(4) tu tu feras attention à ça
CL CL V NC PREP PRO

In this section, we describe the model architecture, the training data preprocessing and training configurations.

3.3.1 Model architecture

The FlauBERT model has the same architecture as BERT (Devlin et al., 2018). It consists of a multi-layer bidirectional Transformer. In our case we use FlauBERT LARGE, which has the following architecture : L = 24, H = 1024, A = 16, where L, H and A respectively denote the number of Transformer blocks, the hidden size, and the number of self attention heads.
The model is relatively simple, with all of the complicated parts contained inside the FlauBERT module, which we do not have to worry about. In our case , we can think of the FlauBERT as an embedding layer and all we do is to add a linear layer on top of these embeddings to predict the tag for each token in the input sequence and this is the concept of fine-tuning a pre-trained transformer model on our use case .

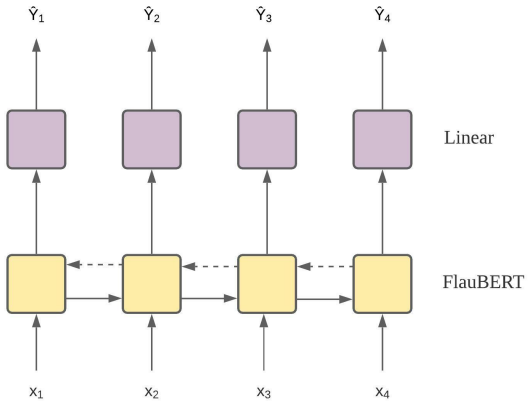

FIGURE 3.20: Fine-tuned Model architecture.

Previously, the yellow squares were the embeddings provided by the embedding layer, but now they are embeddings provided by the pretrained FlauBERT model. All inputs are passed to FlauBERT at the same time. The arrows between the FlauBERT embeddings indicate that FlauBERT does not calculate embeddings for each token individually, but that the embeddings are actually based on the other tokens within the sequence. We say that the embeddings are contextualized.
One thing to note is that we do not define an embedding_dim for our model, because it is the size of the output of the pretrained FlauBERT model and we cannot change it. Thus, we simply get the embedding_dim from the model's hidden_size attribute.

3.3.2 Training data preprocessing

The data used for training and testing the model is based on the output results of the previous method (section 3.2) because there is no existing corpus that contains relevant data for this task. In addition , building a golden corpus manually requires a lot of effort and is time consuming.

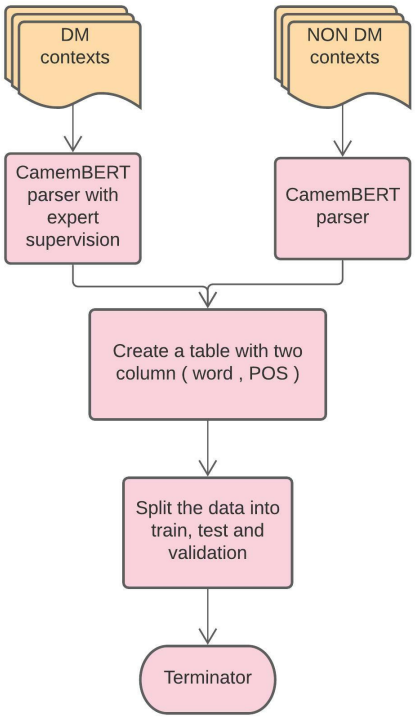

FIGURE 3.21: The process of preparing the data used by the pre-trained model.

3.3.3 Training configurations

For the training hyperparameters configuration, we define our optimizer by following (Liu et al., 2019) and use the Adam optimizer (Kingma and Ba, 2014) with the following parameters:

- Usually when fine-tuning, we use a lower learning rate than normal, this is because we do not want to drastically change the parameters, as it may cause our model to forget what it has learned. This phenomenon is called *catastrophic forgetting*. We pick $5e^{-5}$ (0.00005) as it is one of the three values recommended in the BERT paper.
- We used dropouts with 0.25 value.
- We defined a loss function which is cross entropy while making sure to ignore losses whenever the target tag is a padding token.
- Finally, we used a batch of size 8 and a number of epochs equal to 50.

3.4 KNN algorithm with BERT word embedding

In this section, we introduce another method based on the strategy of K Nearest Neighbor (KNN) with using BERT word embedding to represent the DM under study.
Bert word embedding vectors can represent different levels of token information and capture obvious differences such as polysemy and others. In short, word embedding is capable of capturing the context of a word in a document, the semantic and syntactic similarities, the relation with other words, etc. Word embeddings are mostly used as input features for other models built for custom tasks.
K-Nearest Neighbours (K-NN, KNN, or nearest KN method): K-NN is a standard classification technique that is solely dependent on the classification measure used. It's "non parametric" (only K has to be fixed), and it's purely dependent on the training data.
The concept is as follows: using a labeled database, one may estimate the class of new data by examining the majority class of nearby data (hence the name of the algorithm). The only constant is K, the number of neighbors to take into account. The standard distance is the most often used measure.

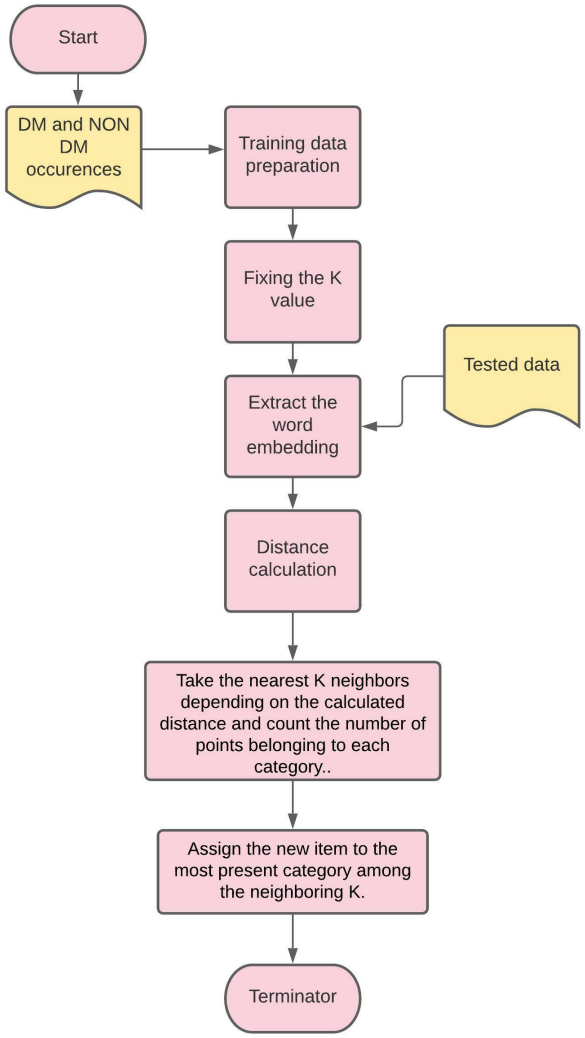

FIGURE 3.22: KNN algorithm using word embedding pipeline..

3.4.1 Training data preparation

As presented in figure 3.22, we used the results provided by method 3.2, due to the good performance achieved and validated by the experts.
We didn't use all of the data since there are so many repeated patterns and environments; instead, we chose the most frequent environment to guarantee that we covered a majority of the different types of DMs environments that exist in the spoken corpora.
Based on this, we defined a set of environments in which each potential DM acts as a DM and others in which they function as a descriptive content component. We have two clusters for each DM in the end, the first called DM and the second called NON DM.

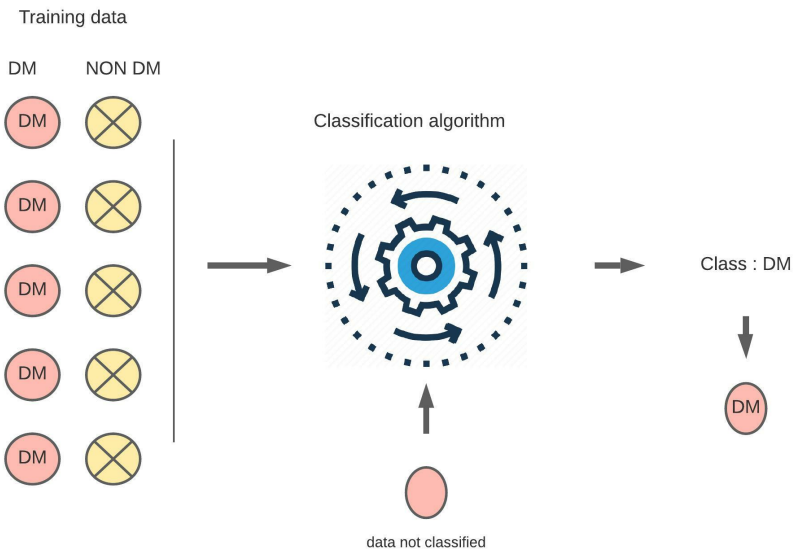

FIGURE 3.23: The structure of training data on the system.

3.4.2 Find the K value

The only hyper-parameter in this method is the k (or K) value, and the most frequent technique to determine it accurately is to run many tests or scenarios and compare the outcomes.
Finally, one takes the k value that produces reasonable accuracy and use it to enhance the system's performance. In our situation, we set the k value to 5 after doing several value testing trials.

3.4.3 Extract the word embedding

The process is intuitively understandable. The first step is to configure the FlauBERT framework and in this case we used the FlauBERTlarge to get a better presentation and cover the unknown words also. The FlauBERT will transform sentences to a list of tokens and try to assign a word embedding vector to each token based on its left and right context. The second step is placing the vector that represents the DM word beside its sentence in the training table. The last is to apply the same second step for all DMs in the occurrences of the spoken corpus to prepare them for the distance calculation with the training data.

3.4.4 Distance calculation

Once we have vectors of the given DM based on its environment, to compute the similarity between generated vectors, statistical methods for the vector similarity can be used. Such techniques are cosine similarity, Euclidean distance, word mover's distance. Cosine similarity is a widely used technique for text similarity. For our case study, we have used cosine similarity (Figure 3.24).
The process is to loop over the training data and calculate the cosine similarity between each occurrence in the training data and the input data. In short , we followed the following steps :

For each example in the training data:

- Calculate the distance between our input and the current iterative observation of the loop from the data;
- Add the distance and index (cluster name) of the observation concerned to an ordered collection of data;

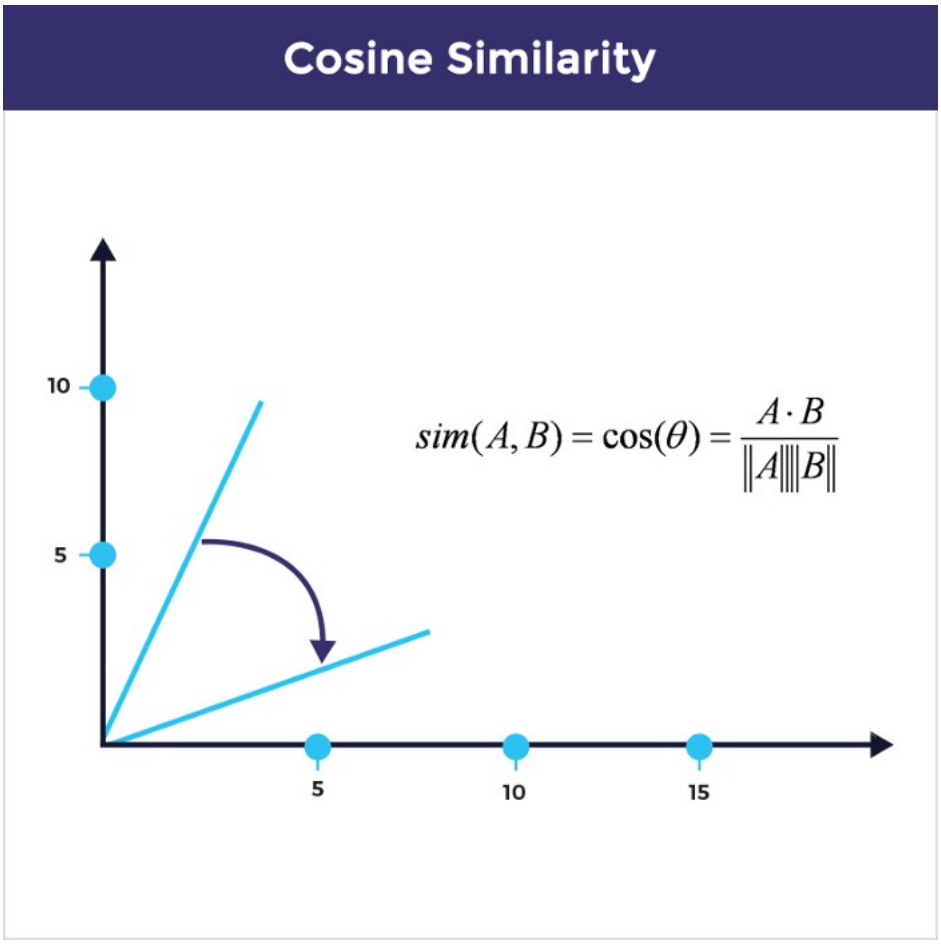

FIGURE 3.24: Cosine similarity equation and illustration.

3.4.5 Classification process

The classification process will go as follow :

- Sort this ordered collection containing distances and indices from the largest distance to the smallest (in descending order).;
- Select the first K's input from the sorted data collection (equivalent to the nearest KNs);
- Get the labels of the selected entries (DM or NON DM);
- Do that for all corpora;

Chapter 4

Evaluation and Discussion

In this section, we describe the spoken corpora used in this study, the text preprocessing pipeline, the results obtained for each experiment and finally we summary these results.

4.1 Data

Data used here consist of 3 French corpora taken from different sources, covering diverse topics and writing styles. The choice of those corpus is based on: (1) the frequency of DM used by their speakers; (2) the writing styles that help us to observe different contexts and words.

CORPAIX is a corpus of spoken French. It was created in the 1970s under the guidance of Jean Stéfanini and then Claire Blanche-Benveniste with the aim of collecting representative spoken French data that could be described as "wide and current use." Since the 90s, all transcripts have been entered on a computer and converted to a standard format containing around 1 million words at the start of the new millennium (Pallaud and Henry, 2004).

Number of Tokens	segments	Speakers
941624	40241	33

TABLE 4.1: Statistics on the CORPAIX corpus.

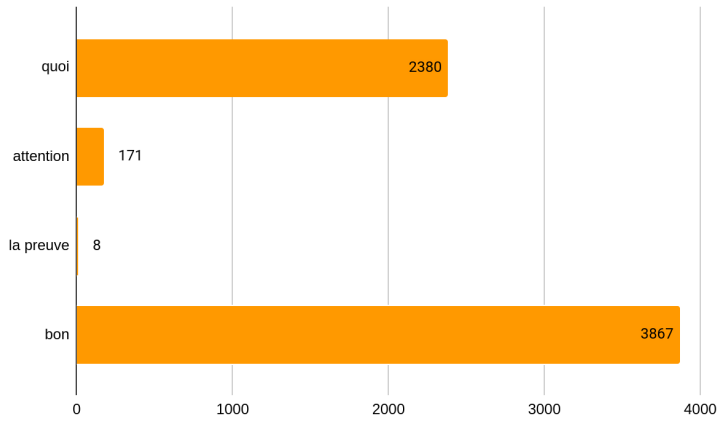

FIGURE 4.1: deployment of the DM in the CorpAix corpus.

ESLO [1]is a sociolinguistics project of the Laboratoire Ligérien de Linguistique of the University of Orleans. Its goal is to create a French spoken corpus (orderly collection of speech recordings). The original version of ESLO, known as ESLO 1, was created by a group of English academics who set out to gather sound materials in Orleans with the pedagogic goal of teaching French as a foreign language in the English public education system.
ESLO 1 contains about 200 interviews with sociolinguistic metadata (speakers and circumstances properties), totaling to over 300 hours of speech, including face-to-face and telephone interviews, public meetings, commercial transactions, family meals, medico-educational interviews, and so on. The second version (ESLO 2) was designed with the goal of gathering approximately 400 hours of audio documents.
The version used now and for this study is a concatenation of ESLO 1 and ESLO 2 to form a collection of 700 hours of recording.

Number of Tokens	Segments	Speakers
649081	53772	28

TABLE 4.2: Statistics on the ESLO corpus.

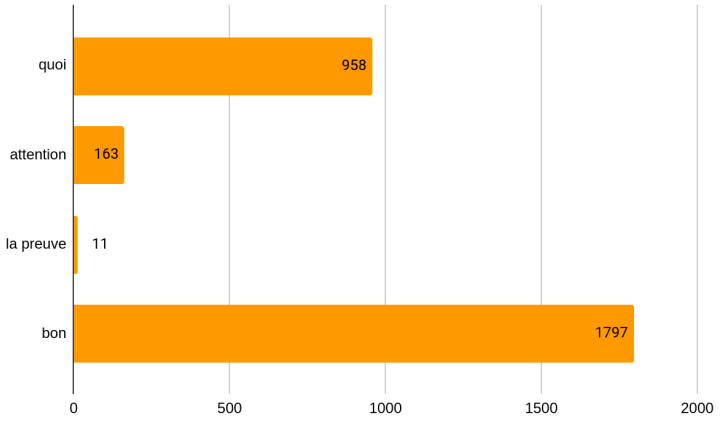

FIGURE 4.2: deployment of the DM in the ESLO corpus.

TCOF [2] (Traitement de Corpus Oraux en Français) is a project born from the desire to keep French spoken data collected in the 1980s for personal research purposes. The team (located at the ATILF laboratory) has developed the architecture of a first corpus database, aligning text and sound with the *transcriber* software. It was gradually enriched from the 2000s by the collaboration of others (teachers-) researchers, ITAs and students of the University of Nancy 2.
TCOF has two major categories: (1) adult-child interaction recordings (children up to 7 years); (2) adult interaction recordings. Both are of various durations: 5 to 45 minutes or more. In our study, we used 51 documents from the adult interaction recordings.

[1] http://eslo.huma-num.fr/
[2] https://www.atilf.fr/ressources/tcof/

Number of Tokens	Segments
149292	19527

TABLE 4.3: Statistics on the TCOF corpus.

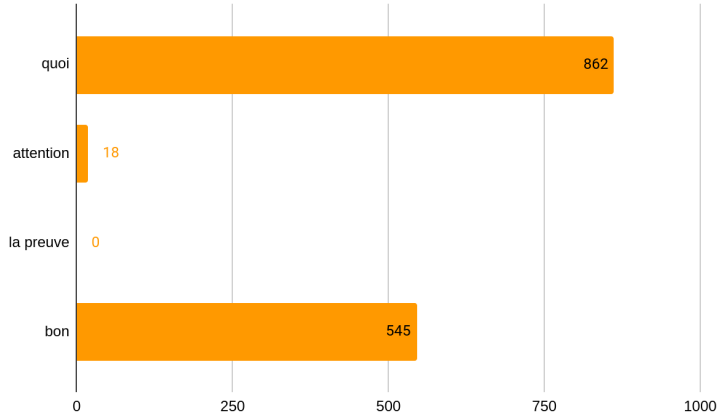

FIGURE 4.3: deployment of the DM in the TCOF corpus.

We used the *selectolax* tool to extract the text from the TCOF corpus. For the CORPAIX and ESLO corpora, we download them directly from their websites. The total size of the text before preprocessing is 8.0 MB.
After analysing the data, we applied a cleaning and normalization procedure for all corpus in order to make them compatible with the parser and to avoid ambiguities:

- We add a space in order to split concatenated words and punctuation.
- We add a space after the apostrophe.
- We split segments that had more than 500 words, keeping the same speaker.
- All the data were Unicode-normalized in a consistent way (UTF-8).

4.2 Experiments and results

In this section, we present how we applied the previously mentioned methodologies on our date and illustrate their achieved results in terms of precision, recall and F1-score. This part will be divided into four subsections. Each of them focuses on the discussion of the experiment that took place, and on the findings obtained using one method.

4.2.1 Unitex with internal and external resources

We applied this method on the data extracted from the three spoken corpora mentioned in section 3.1. First, we use the Unitex platform resources. Second, we use external resources and avoid internal ones. The experiment was done with two objectives: (1) to evaluate the method's performance on the task of DM identification; (2) to evaluate the performance and the usefulness of Unitex resources.

Corpus	Precision	Recall	F1-score
CORPAIX	0.91	0.49	0.63

TABLE 4.4: Evaluation of Unitex with its internal resources on the identification of marker *bon* in 100 samples from the CORPAIX data.

Corpus	Precision	Recall	F1-score
CORPAIX	1	0.49	0.65

TABLE 4.5: Evaluation of Unitex with external resources on the identification of marker *bon*in 100 samples from the CORPAIX data.

Results

Tables 4.4 and 4.5 present respectively the final evaluation on the 100 excerpts from the CORPAIX data for the first and the second scenarios using Unitex. The results highlight an important difference between the scenarios, which is the precision of the resources. External resources outperform Unitex's resources by a substantial margin. Unitex with external resources seems to perform moderately better than Unitex with its internal resources in the CORPAIX datasets. In addition, the results on the two remaining dataset (from TCOF and ESLO) seem to be the same. The recall measure is the same in both scenarios.

4.2.2 Syntactic and lexical patterns

The experiment for this method is almost identical to the previous one, except that the distribution of the evaluated sampling for each DM is not the same due to the imbalanced data.

DM	TCOF	CORPAIX	ESLO
attention	14	51	37
bon	349	63	119
la preuve	0	6	0
quoi	207	90	72

TABLE 4.6: Distribution of the evaluated DM in the three corpora.

Corpus	Samples number	Token number
CORPAIX	613	5706
ESLO	228	1664
TCOF	167	1179

TABLE 4.7: Statistics about the evaluated data.

DM	Precision	Recall	F1-score
attention	0.62	0.94	0.74
bon	0.98	0.92	0.94
la preuve	0.8	1	0.88
quoi	0.95	0.88	0.91

TABLE 4.8: Results obtained from the CORPAIX data.

DM	Precision	Recall	F1-score
attention	0.31	1	0.47
bon	0.96	0.91	0.93
la preuve	-	-	-
quoi	0.93	0.84	0.88

TABLE 4.9: Results obtained from the ESLO data.

DM	Precision	Recall	F1-score
attention	0.67	0.67	0.67
bon	0.95	0.76	0.84
la preuve	-	-	-
quoi	0.94	0.85	0.89

TABLE 4.10: Results obtained from the TCOF data.

Results

The final evaluation for each DM in different corpora is reported in Table 4.8 , 4.9 and 4.10. One can observe that this method works well and is stable in terms of identification of *quoi*, *bon* and *la preuve*. Also, the identification of *attention* in CORPAIX and TCOF data is slightly better than the identification in ESLO. This could be attributed to the data characteristics. Finally, based on the precision value of *attention* over the three corpora, we can observe that the method's sensitivity over the positive context of *attention* is very low, which leads the method to consider some non-DM as DM.

4.2.3 Fine-tuning pretrained Bert model for POS tagging task

For the model architecture, we used the same architecture as the one described in section 3.3. For the hyper-parameter, we used a learning rate of 0.00005, with a batch size equal to 8 and 50 epochs for training.
When we trained the model on POS labeling task, we used a new category, called DM, to discriminate between discourse markers functionality and other grammatical functions. The POS for each word given in its context was identified with the CamemBERT parser. Our data was taken from the three corpora and splitted into train (90%), development (5%) and test (5%), as illustrated in table 4.11.
During the training, we selected the best model version according to the development accuracy.

train	dev.	test
668	35	35

TABLE 4.11: Distribution of samples in train, dev and test data for DM *attention* and *bon*.

DM	Precision	Recall	F1-score
Attention	1	0.16	0,27
Bon	1	0.33	0.49

TABLE 4.12: Results of the pre-trained model for DM *attention* and *bon*.

Results

For the pre-trained model, the results are reported in 4.12. The precision is pretty good for both DMs even for low quantity samples for ' attention . The recall is very low in general. One can observe that the quantity of 'bon' samples increment the recall a little bit compared to ' attention'.

4.2.4 KNN algorithm with BERT word embedding

The setup of this method is the same as the one described in section 3.4. The training examples are vectors obtained from the FlauBERT large pre-trained model on French language, each with a class label (DM or non-DM).
The training phase of the algorithm consists only in storing the feature vectors and class labels of the training samples (13 vectors for DM context and 13 for non-DM). After several testing, k is set to 5, and an unlabeled vector, about which we want to know whether it works as a DM or not, is categorized by assigning the label that appears most frequently among the k training examples closest to that input.

Corpus	Samples number	Token number
CORPAIX	123	861
ESLO	69	483
TCOF	11	77

TABLE 4.13: Statistics about the evaluated data in order to classify the DM *attention* in the three datasets.

Corpus	Precision	Recall	F1-score
CORPAIX	0.88	0.88	0.88
ESLO	0.89	0.88	0.88
TCOF	1	0.5	0.66

TABLE 4.14: Results obtained for the categorization of DM *attention*.

Results

The results are reported in Table 4.14. The best achieved results for the DM *attention* in this study.The KNN algorithm performs better in CORPAIX and ESLO than in TCOF datasets. This is due to the TCOF corpus characteristics and at the same time to the low generality of the training samples. One can observe that KNN in TCOF outperforms KNN in CORPAIX and ESLO, due to the low quantity of data tested.

4.3 Discussion

The experiments of the previous section were carried out in order to identify the DM *quoi*, *attention*, *la preuve* and *bon* in French spoken corpora by taking into account their context in each experiment. It is based on the literature stating that DM are functional-pragmatic rather than morphosyntactic forms. In this section, we discuss the meaning, he importance and the relevance of the outcomes of each applied method. In addition, the section focuses on explaining and evaluating what we found.

4.3.1 Rule-based approach methods

The first scenario from the first method, which exploited the Unitex platform with its own resources (dictionaries and taggers), had some problems in the identification of DM, as observed in table 4.4. This difficulty is due, on the one side, to the internal resources which affected the precision metric and, on the other side, to the low and noisy rules used in the graphs. These rules cannot cover all the positive contexts of the studied DM, which is obvious in the value of recall.
Unitex's resources do not always identify the right grammatical category of tokens. This happens when the context of a token is not taken into account, as illustrated in figure 4.4 for the sentence *tout est bon* and this is because Unitex finds that *est* has an entry of the form 'est,.A+z1' in dictionaries.

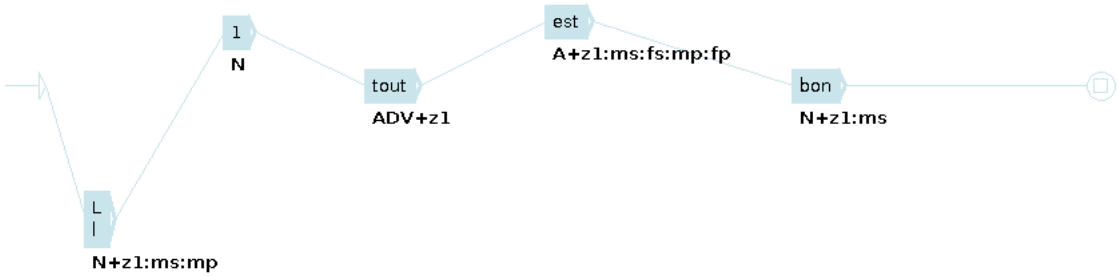

FIGURE 4.4: Example of mislabeling of the grammatical category by Unitex.

Based on the recall value, we observe that using rules for the positive context of the DM affects the performance of the method, because of the difficulty of covering all the positive contexts of DM. The motivation behind using the positive context is that Unitex does not have a negation of the left context in its graphs. Unitex provides negation just for the right context.
There is a way to apply a negation to the grammatical category like using '!V' which will identify all grammatical categories except the verbs but, in our case, we have not just one category in the left context and, if we use this solution, we will also have trouble with precision, because if we apply '!V' and '!N' , we will find the verb in '!N' and nouns in '!V'.

The problem of precision was solved with the second scenario by using the CamemBERT parser, which takes into account the word context before determining the final grammatical category. However, the problem of covering all DM still exists, because of the the use of positive context. When compared to the first method, the second method performs well, leading us to believe that the negative context can be more helpful than the positive context, and that a larger knowledge base can make a significant difference when using a rule-based approach, because it provides us with more information.
The second technique demonstrates that DM identification cannot be accomplished only through the use of syntactic patterns. Using additional lexical information will assist the algorithm in distinguishing between DM and non-DM.
As we can see in table 4.9 , the second method does not achieve good results for the DM *attention* and this is due to the noise that exists in the data, and especially in CORPAIX data, as illustrated in example (3), and also to the variable environments

of *attention*, especially in the ESLO data, as illustrated in example (4). There, *attention* is a non-DM but, since the pattern does not exist in the *Le Monde* corpus, the system identifies the occurrence as a DM. Such cases affect negatively the precision value.

(5) vont faire = attention peut-être de comment
faire plus = attention mais quand =
L3 - et sinon ils euh ils ils font pas trop . attention euh

(6) vos parents faisaient attention à votre façon
qu' ils faisaient attention à la façon

4.3.2 Machine learning approach methods

Based on the outcomes of the KNN algorithm, we observed that using word embedding can identify correctly the function of *attention* in different contexts, owing to the information contained in the embedding vector. As previously stated, word embedding can extract syntactic, lexical, and semantic information from the context, allowing us to determine whether the investigated occurrence is a DM or a descriptive content component.

In the case of the TCOF corpus, we may claim that the findings are influenced by the corpus' characteristics and by a lack of data in the training dataset.

For the pre-trained model, the test was done for the classification of *bon* and *attention*. The model's outcomes were pretty good in terms of precision compared to recall owing to the quantity of training data used. Because of the limited context variety and the balancing training data, the findings of *bon* were better than those of *attention*.

Based on the results of both the KNN algorithm and the pre-trained model, text augmentation is recommended to improve results and methods: (1) by allowing the pre-trained model to see more diverse environments and examples; (2) by increasing the training data and the k value, for making the KNN algorithm more robust.

Chapter 5

Conclusion

DM are key indicators of discourse structure, and have been shown to be useful devices for (a) segmenting discourse, (b) identifying relations between units, and (c) identify also the attitudes and sentiment of the speaker. Identifying DM is often crucial for understanding the communicated message, especially in spoken corpora, and for making the discourse coherent.

The research reported in this document concerns the grammatical functions of some polyfunctional DM such as *attention*, *bon*, *quoi*, *la preuve* in spoken dialogue. It applies rule-based and machine learning methods in order to detect the category of each occurrence in several contexts.

The outcomes obtained after testing the rule-based method, that took into account only the grammatical category of the items that surround the occurrence show that DM cannot be identified only on the basis on the grammatical categories found in its environment. In addition, the occurrence can act as a DM and as a part of descriptive content in the same environment by just by changing the lexical units and by keeping the same grammatical categories.
Following the latter conclusion, our next move was to use both syntactic and lexical constraints, as mentioned in section 3.2. The resulting outcome confirms that the combination of syntactic and lexical information can achieve the goal of recognizing the function of the occurrence (DM or non-DM).

Unfortunately, recognizing the DM *attention* seems to be critical to both strategies. This prompted us to use the word embedding of the occurrence in order to access its syntactic, lexical and semantic information for improving the identification. The KNN algorithm proves this hypothesis by showing that using word embedding can be beneficial in the case of *attention*.

Building a pre-trained model that can recognize and identify the occurrence functionality in discourse is still a crucial objective because it would allow us to get better results than the previous methods. However, in some cases, it is necessary to get more positive data, that is, data in which the occurrence is a DM. Our mentioned pre-trained model gives good results but not as good expected because it is still affected by scarce or unbalanced data, especially for *attention* and *la preuve*.

To conclude, the identification of the polyfunctional occurrences is still a challenge and needs a lot of analysis to determine the better way to deal with each DM because it is obvious that they have different patterns and behavior. The deep learning methods show their ability to detect correctly some DM, on the basis on relatively low resources. However, they need a lot of data to have a significant number of DM and environments in order to cover a lot of situations to adapt to different discourse genres and registers. The work is still continuing and our perspective is to contribute to this task, in particular by extending the analysis to other polyfunctional DM such as *tiens*, *tu parles*, *bref* and *bon Dieu*.

Bibliography

Brinton, Laurel J. (2017). *The evolution of pragmatic markers in English: pathways of change*. Cambridge University Press.

Dargnat, Mathilde (2021). "Les particules énonciatives". In prep. for L'Encyclopédie Grammaticale du Français. URL: http://encyclogram.fr/.

Devlin, Jacob et al. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*. URL: https://arxiv.org/abs/1810.04805.

Dostie, Gaétane (2004). *Pragmaticalisation et marqueurs discursifs. Analyse sémantique et traitement lexicographique*. Bruxelles: De Boeck / Duculot.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*. URL: https://arxiv.org/abs/1412.6980.

Lamiroy, Béatrice and Pierre Swiggers (1991). "The status of imperatives as discourse signals". In: *Suzanne Fleischman and Linda R. Waugh (eds.)* London:Routledge, pp. 120–146.

Le, Hang et al. (2019). "Flaubert: Unsupervised language model pre-training for french". In: *arXiv preprint arXiv:1912.05372* abs/1909.05364. URL: https://arxiv.org/abs/1912.05372.

Liu, Y et al. (2019). "RoBERTa: A robustly optimized bert pretraining approach, 2019". In: *arXiv preprint arXiv:1907.11692* 364. URL: https://arxiv.org/abs/1907.11692.

Martin, Louis et al. (2019). "Camembert: a tasty french language model". In: *arXiv preprint arXiv:1911.03894* abs/1909.05364. URL: https://arxiv.org/abs/1911.03894.

Nie, Allen, Erin Bennett, and Noah Goodman (2019). "DisSent: Learning Sentence Representations from Explicit Discourse Relations". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy). Ed. by editor. Association for Computational Linguistics, pp. 4497–4510. URL: https://aclanthology.org/P19-1442.

Pallaud, Berthille and Sandrine Henry (2004). "Amorces de mots et répétitions: des hésitations plus que des erreurs en français parlé". In: *7es Journées Internationales d'Analyse Statistique des Données Textuelles*, pp. 848–858.

Petukhova, Volha and Harry Bunt (2009). "Towards a multidimensional semantics of discourse markers in spoken dialogue". In: *Proceedings of the Eight International Conference on Computational Semantics*, pp. 157–168.

Sileo, Damien et al. (2020). "DiscSense: Automated Semantic Analysis of Discourse Markers". In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 991–999. URL: https://aclanthology.org/2020.lrec-1.125.

Stede, Manfred (2012). *Discourse Processing*. Morgan & Claypool.

Wu, Xing et al. (2020). "TransSent: Towards Generation of Structured Sentences with Discourse Marker". In: *CoRR* abs/1909.05364. URL: http://arxiv.org/abs/1909.05364.

Zufferey, Sandrine and Andrei Popescu-Belis (2004). "Towards Automatic Identification of Discourse Markers in Dialogs: The Case of *Like*". In: *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*. Ed. by editor. Association for Computational Linguistics, pp. 63–71. URL: https://aclanthology.org/W04-2313.