An evaluation of part-of-speech taggers for French

Mattiuzzi, Silvia

UNIVERSITÉ DE GENÈVE
FACULTÉ DE TRADUCTION ET D'INTERPRÉTATION

Silvia Mattiuzzi

An evaluation of part-of-speech taggers for French

Directrice : Pierrette BOUILLON
Jurée : Johanna GERLACH

Mémoire présenté à la Faculté de traduction et d'interprétation (Département de traduction, Unité d'Italien) pour l'obtention de la Maîtrise universitaire en traduction et technologie (MATT)

Année académique 2020-2021 / Session d'août

UNIVERSITÉ DE GENÈVE
FACULTÉ DE TRADUCTION ET D'INTERPRÉTATION

Déclaration attestant le caractère original du travail effectué

J'affirme avoir pris connaissance des documents d'information et de prévention du plagiat émis par l'Université de Genève et la Faculté de traduction et d'interprétation (notamment la *Directive en matière de plagiat des étudiant-e-s, le Règlement d'études des Maîtrises universitaires en traduction et du Certificat complémentaire en traduction de la Faculté de traduction et d'interprétation ainsi que l'Aide-mémoire à l'intention des étudiants préparant un mémoire de Ma en traduction*).

J'atteste que ce travail est le fruit d'un travail personnel et a été rédigé de manière autonome.

Je déclare que toutes les sources d'information utilisées sont citées de manière complète et précise, y compris les sources sur Internet.

Je suis conscient-e que le fait de ne pas citer une source ou de ne pas la citer correctement est constitutif de plagiat et que le plagiat est considéré comme une faute grave au sein de l'Université, passible de sanctions.

Au vu de ce qui précède, je déclare sur l'honneur que le présent travail est original.

Nom et prénom : Mattiuzzi Silvia

Vicenza, le 12 septembre 2021

Silvia Mattiuzzi

Abstract

Annotated corpora are widely employed in a variety of fields such as linguistics, translation studies, natural language processing, etc., and part-of-speech tagging is one of the most common form of corpus annotation. This master thesis presents an evaluation of three part-of-speech taggers for French. All systems are freely available for non-commercial use but differ in the approach to POS tagging (a MaxEnt Markov model, a probabilistic model with decision trees and an artificial neural network model) as well as in the way they interface to the user. Two series of experiments are carried out where taggers are tested without training or tuning: in the first series, the procedures necessary for the deployment of each tagger are illustrated so as to ascertain that their implementation is simple enough for individuals with moderate knowledge in computer science. A feature inspection is also performed to give account of the modules embedded in each system (tokenization, lemmatisation, etc.) and the file formats that can be handled. In the second series of experiments, a quantitative evaluation of the taggers' performance on four different text types (speech transcripts, literature, product reviews and legal texts) is provided: a black-box usage is simulated to identify which system produces the most accurate annotation for each text typology. The aim of this study is to provide users with an overview of different alternatives for the morphosyntactic annotation of French corpora and the opportunity to choose the POS tagger that best suits their needs – whether it is in terms of the quality of the annotation with respect to the text typology, the format of the files to be processed or the skills required to deploy it.

Table of Contents

Introduction

The introduction of machine learning algorithms in the 1980s ushered in the "revolution" of Natural Language Processing. Since then, a lot of effort has been put into designing and improving computational methods to process electronic texts of spoken and written form of human communication. The main task of NLP can be summarized as the processing of an unstructured text to produce a representation of its meaning (Singh, 2018). But the processes involved in the NLP pipeline are multiple and operate on data from different angles. At the syntactic level, the smallest and meaningful building blocks of texts, namely words, are identified along with the role they play in a sentence. At the semantic level, it is the meaning of these words that is determined; while at the pragmatic level, the context is exploited to provide the most suitable and meaningful interpretation (Assal, 2011). Part-of-speech tagging is an NLP process which operates at the morphosyntactic level: it takes a sentence or a text as input and returns as output the same sentence or text where every linguistic unit has been labelled with a part of speech.

In the past three decades, a considerable number of computational linguists and enthusiast programmers have ventured into building codes to carry out this specific sequence labelling task. As a result, several approaches and implementations have been brought forth, especially but not exclusively in the academic field. Nevertheless, only some of these part-of-speech tagging systems are open source and freely available. But even when they are, one cannot confidently assert to be able to run them. As a matter of fact, most of these codes are not compiled into a software or provided with a graphical user interface, thus a fair knowledge of programming is required to deploy them, one that researchers and students without a solid background in computer science or computational linguistics may not have.

Besides, when confronted to the choice of a part-of-speech tagger for the annotation of a corpus of texts, usability may not be the only factor one wishes to consider. The multitude of diverse content available nowadays along with the need to improve existing solutions to language-specific issue keep fostering the creation of new systems which exploits different techniques for POS tagging or the adaptation of the existing ones. Recent researches for French have investigated different strategies to deal with unconventional text typologies, such as user generated content (Nooralahzadeh et al., 2014), and with language variation, for example Schmid (2019) and Magistry et al. (2019). It appears that

some systems that are also provided with hand-annotated data and other lexical resources are more accurate than others for certain text types. For example, the linear-chain conditional random fields (CRFs) model used to annotate French social media data (Nooralahzadeh et al., 2014) achieves a 91,9% accuracy. This is made possible because it is easier to incorporate a lot of features when using a discriminative sequence model. Schmid (2019) presents, instead, a tagger based on recurrent neural nets which processes the character sequences of words with a bidirectional Long Short-Term Memory network (BiLSTM). The tagger is capable of learning spelling variations, which is a distinctive trait of historical texts, and the tagging accuracy improves to 96,28%. Unconventional text typologies are therefore a challenge.

At last, this master thesis found its grounds on my recent involvement in the construction of CHEU-lex[1], a parallel and comparable trilingual corpus of legal texts. This corpus has been developed to study the impact of the European linguistic diversity (Eurolects) in the Swiss federal legislation produced at national and international level. In the framework of this project, which have seen the participation of some translation students, I was required to perform the automatic part-of-speech annotation of its French sub-corpora, among other tasks. The software initially appointed for the task was making systematic errors in the POS annotation of French contractions, in particular, the ones containing punctuation marks such as apostrophes. Since it was not possible to fix or bypass the issue, I was suggested to try a different tagger which was ultimately chosen to carry out the annotation task. Following the automatic tagging stage, a manual consistency checking and correction of the labelled output was carried out. Although being a necessary step for this type of work and a time-consuming one as every industry insider is well-aware, the amount of manual work expended for corrections arose doubts, in retrospect, as to which software was genuinely the best suited to perform the task for the type of corpus in hand.

The above-mentioned reasons are therefore the driving forces of this study, which aims to collect various morpho-syntactical sequence labellers and provide an evaluation of their performance and use. To do so, two series of experiments are conducted without the training or tuning of the taggers, hence a first-time approach to POS tagging is simulated: initially, the selected systems are run to provide an overview on the

[1] https://transius.unige.ch/en/research/cheu-lex/

requirements for their deployment, the file formats accepted, and the different modules embedded within its architecture. A brief account of the performance of these modules (tokenization and lemmatisation) on particularly challenging linguistic phenomena is also provided. Finally, a black-box usage is simulated, and the accuracy of the POS annotation is computed for each system on different text typologies.

Research Question

This research focuses on three part-of-speech tagging approaches for French. All systems considered are freely available for academic use, have a language model for the annotation of French corpora and are assumed to be relatively easy to deploy from the perspective of translators, students, or other researchers, who do not have an in-depth knowledge of programming and need a quasi-straightforward system to annotate a text or a corpus.

The aim of this research is twofold: on the one hand, there is the desire to identify which software produces the best annotation for a specific type of text and, on the other hand, there is the concern as to which system is the easiest to use by individuals who do not have an extensive knowledge in computer science. The usability of the systems is evaluated in terms of deployment, integrated functionalities, and file formats handled.

The questions to be answered are the following: how is the tagging performance affected by a specific text typology and/or well-known linguistic challenges? Which system's interface is more user-friendly? And finally, are the taggers provided with embedded modules that deals with various file formats and covers all stages of text pre-processing and processing?

Independent variables	**Dependent variables**
Text typology	Correct labels
User's skills	System's interface
Systems' architecture	Modules and File formats

Table 1. Dependent and Independent Variables

The rest of this thesis is structured as follows: in chapter 1, I will introduce part-of-speech tagging, discuss the processes that are related to it as well as some of the main strategies used to deal with this sequence labelling task. Existing methodologies and metrics for the

evaluation of language software and of part-of-speech taggers are outlined in chapter 2. Chapter 3 presents information on the evaluation's framework of this thesis and the experimental setup: the methodology and metrics applied are given in 3.1, the corpora used for the experiments are described in 3.2 and 3.3; finally, the selected taggers are presented in 3.4. An account of the two series of experiments which constitute the present evaluation is given in chapter 4 while the results will be presented and discussed in Chapter 5.

1. Part-of-Speech Tagging

Part-of-speech tagging is a language processing task involving "the identification of the morphosyntactic class of each word form using lexical and contextual information" (Paroubek, 2007, p.99). It is an example of how information at the morphosyntactic level of linguistic description can be input to a corpus. A linguistic description of this kind can prove beneficial during the linguistic analysis of corpora. In this chapter, I first present this type of annotation by taking a closer look at the two concepts of "parts of speech" and "tagging" (1.1). Then, I discuss the other tasks, or modules, that are usually embedded in tagging systems, such as text segmentation (1.2.1) and lemmatisation (1.2.2). Finally, I present in general terms some of the most common approaches to POS tagging.

1.1 Task definition

According to Schachter (1985), all natural languages make parts-of-speech distinctions, even though there is a significant variation concerning their typology and number across different languages. Therefore, having a tool that can annotate parts of speech in a text or a corpus, so as to enable an investigation of both well-known and unfamiliar languages on a common ground, would be of great value and interest.

Parts of speech, also known as POS, are labels which encode information about the grammatical nature of words. Jurafsky and Martin (2020, Chapter 8, p.2) argue that "while word classes do have semantic tendencies […] parts of speech are defined instead based on their grammatical relationship with neighbouring words or the morphological properties about their affixes." Parts of speech are generally grouped into two broad categories, namely open and closed classes (Jurafsky & Martin, 2020). Adjectives, adverbs, common and proper nouns, and most verbs fall into the open class category (Table 2) and are generally defined as content or lexical words. The smaller group of interjections also belongs to this category. The term "open" refers to the fact that due to different linguistic phenomena, such as neologism and calques, new coined entries are quite common in this class. As far as open and closed classes are concerned, Gil (2000, p.182) argues that "not all syntactic categories are of equal size", because "open syntactic categories are productive, and contain a large, sometimes infinite number of members, whereas closed syntactic categories are non-productive, generally consisting of a small number of members".

	POS tag	Description	French Example
OPEN CLASS	ADJ	Adjectives are nouns modifiers that specify their properties or attributes. French adjectives agree in gender and number with the noun they modify (both in attributive and predicative position).	beau|bel, beaux, belle, belles (= beautiful) idéal, idéals, idéale, idéales (= ideal)
	ADV	Adverbs typically modify adjectives, verbs or other adverbs for categories such as time, place, direction or manner.	assez facile (= rather simple) faire vite (= to be quick)
	INTJ	Interjections are words or expressions occuring as a single utterance which express a spontaneous feeling or reaction.	bof (= disdain) hélas (= pain, regret)
	NOUN	Nouns are a part of speech typically denoting a person, place, thing, animal or idea.	fils (= boy|son) abeille (= bee) beauté (= beauty) voiture (= car)
	PROPN	A proper noun is used to name specific (i.e. opposed to generic), one-of-a-kind individuals, places, or objects.	L'Oreal Paris ONU Charles-Louis de Secondat
	VERB	Verbs are used to espress an event, a physical action, a mental action, or a state of being. This tag is reserved for full lexical verbs.	Il pleut. Elle a couru vers le train. Je me demande ce qui arrivera. Ils sont tristes!

Table 2. Open class. Parts of speech from the Universal Dependencies Tagset.

Articles, conjunctions, prepositions, pronouns, auxiliary and modal verbs are grouped into the closed class category (Tables 3 and 4). All the items falling into this class are relatively fixed in number as new coinages are quite rare. Parts of speech in the closed class are defined as function or grammatical words because they often have structuring uses in grammar. Numbers and particles also belong to this group, although the latter are more abundant in languages like English rather than French.

	POS tag	Description	French Example
CLOSED CLASS (1st part)	ADP	Adposition is a term that icludes both prepositions and postpositions. French has only prepositions.	pour, à, au-dessus, depuis
	AUX	An auxiliary verb accompanies the lexical verb (VERB) and expresses grammatical distinctions which are not carried by it, such as person, number, tense, mood, aspect, and voice. French auxiliary verbs can be divided into tense auxiliaries, modal auxiliaries and passive auxiliaries.	Les témoignages sont recueillis par la police. (to be as passive auxiliary) J'ai réussi l'examen. (to have) On voudrait boire du café. (to will)
	CCONJ	A coordinating conjunction is used to create links between words or larger constituents of equal grammatical rank and syntactic importance.	mais (= but), ou (= or) et (= and), donc (= thus)
	DET	A determiner is word that introduces a noun or gives information about the quantity of a noun or clarify what the noun refers to. This category includes possessive, demonstrative, interrogative, relative and quantity determiners as well as articles.	articles: les filles, un homme possessive det.: ton oncle demonstrative det.: ce vélo quantity det.: tous les matins

Table 3. Closed class (1). Parts of speech from the Universal Dependencies Tagset.

	POS tag	Description	French Example
CLOSED CLASS (2nd part)	NUM	A numeral is a word that expresses a number and a relation to the number, such as quantity, sequence, frequency or fraction. It may includes both cardinal and ordinal numbers.	un, 2, XVII, quatrième, 6ème, 1⁄2
	PART	Particles are function words that must be associated with another word or phrase to impart meaning. They do not satisfy definitions of other universal parts of speech.	Que pense-t-il ? (= What does it think?)
	PRON	Pronouns are grammatical words that represent a noun or a noun phrase of specific meaning, already used elsewhere in the context. They can also play the role of an absent noun.	personal pron.: je, nous, elles .. demonstrative pron.: celui .. reflexive pron.: te, se .. interrog./relative pron.: qui, que
	SCONJ	A subordinating conjunction is a word or phrase that links a dependent clause, which cannot stand alone as a complete sentence, to a main clause.	quand (= when) parce que (= because)

Table 4. Closed class (2). Parts of speech from the Universal Dependencies Tagset.

Finally, few more elements (Table 5) are usually present in texts but are excluded from the abovementioned categorisations: punctuation marks, foreign expressions, symbols, and abbreviations. In some cases, one or more *residual* or *miscellaneous* part of speech label is devised for the encoding of these elements that do not fit into the binary classification "open-closed" (Cloeren, 1999).

	POS tag	Description	French Example
OTHER	PUNCT	Punctuation marks are non-alphabetical characters and character groups used to delimit linguistic units in printed text.	, . : ; / ? ! ( )
	SYM	A symbol is a word-like entity. This part-of-speech usually include symbols, emoji, email or website addresses, and so on.	$, %, +, −, ×, ÷, =, <, > :), 😉 https://www.oed.com/
	X	Everything that does not fall into the other part-of-speech category. It could be foreign words, non-words, etc.	

Table 5. Other parts of speech from the Universal Dependencies Tagset.

Tables 2 to 5 show the 17 parts of speech that compose the tagset adopted by the Universal Dependencies (UD) framework (Nivre et al., 2016). It should be noted that this is just an example of a tagset. To the present day, there is no universal agreement on part-of-speech labels, even though several attempts have been made to arrive at a common POS annotation system for all the languages of the world. The Universal Dependencies (UD) project represents precisely one of these attempts as well as the EAGLES' recommendations for the morphosyntactic annotation of corpora (Leech & Wilson 1996).

The term “tagging” is defined as the action aimed at attaching a descriptor, or a label, to someone or something in order to identify it. Part-of-speech tagging is a natural language process for which “we assign to each word x_i in an input word sentence, a label y_i, so that the output sequence Y has the same length as the input sequence X” (Jurafsky & Martin, 2020, Chapter 8, p.1). It is, therefore, a sequence labelling task, where POS tags encode morphosyntactic information of each one of the linguistic units forming a sentence in a specific language, providing an overall representation of it.

Part-of-speech tagging is one of the most popular and well-established type of linguistic annotation since it is now possible to tag large amount of text with a relatively high accuracy; it is also an often-preliminary stage to other activities in Natural Language Processing, such as syntactic parsing (Leech & Smith, 1999) and semantic analysis. Part-of-speech taggers generally operate by taking an input file, usually a plain text or an XML file, containing few sentences, a text, or a corpus, and returning an output where each word is appended to a POS tag. Sometimes the POS tag is also accompanied by the canonical form of the word, known as lemma, and, in the case of the most comprehensive systems, even by other morphological features, such as genre, number, etc. Two examples of the POS annotation of the French sentence “*Je l’entendais à peine*” (= I could hardly hear him) are provided below:

(1) annotation with part-of-speech tags and lemmas:

Token	POS	Lemma
Je	*PRO:PER*	*je*
l'	*PRO:PER*	*la|le*
entendais	*VER:impf*	*entendre*
à	*PRP*	*à*
peine	*NOM*	*peine*

(2) annotation with lemmas, part-of-speech tags, and lexical/grammatical features:

Token	Lemma	POS	Morphological features
Je	il	PRON	Number=Sing|Person=1|PronType=Prs
l'	le	PRON	Number=Sing|Person=3|PronType=Prs
entendais	entendre	VERB	Mood=Ind|Number=Sing|Person=1|Tense=Imp|VerbForm=Fin
à	à	ADP	_
peine	peine	NOUN	Gender=Fem|Number=Sing

The layout of the input file depends on the modus operandi of the POS taggers, and thereby on the algorithms included in its implementation; the same can be said for the text generated as output. As a matter of fact, some POS taggers require input texts to be already segmented into phrases or tokens; others successfully run input files where sentence boundaries are not delimited by tags such as <s> and </s> but rather, make use of punctuation marks and control character, for example the carriage return, present in the text. In the same way, the output file layout varies greatly: words may be arranged in one-token per-line with the POS tags, lemma and morphological (or lexical) features separated by a tab or POS tags may be added in text and attached after the token by means of an underscore "_" or a slash "/".

1.2 Related tasks

Different modules are usually embedded within the architecture of a POS tagging system: at the pre-processing stage we may find a module that operates sentence splitting or a tokenizer that segments sentences into tokens (tokenization); both of these modules deal with the task of segmenting a text even though at different levels. At the processing stage, instead, there is usually a module for tagging tokens with parts of speech/morphological features and one that operate lemmatisation, called lemmatiser. However, not all of these modules are present in every POS tagger. Furthermore, electronic texts are stored in a variety of formats, character sets, typing convention and layouts which are not always supported by part-of-speech tagging systems. In some cases, the modules performing text segmentation are absent, hence it could be necessary to plan a pre-processing phase in which the input text is adjusted to the requirements of the system being used.

One of the first steps in the adaptation of a text for processing reasons is usually normalization, that is, the input text is converted into a specific encoding standard (the most widely used being UTF-8). During normalization, text formatting can be lost, thus it is also recommended to encode any relevant information as SGML or XML markup. This other step, which entails the annotation of a text with structural markup, is of particular importance since some formatting information may be useful for the core step of the pre-processing phase which is text segmentation.

Text segmentation is discussed in more detail in the following section 1.2.1 while section 1.2.2 covers lemmatisation – another important task that is usually carried out by the lemmatiser embedded within the architecture of the POS tagging system.

1.2.1 Text segmentation

The segmentation of a text is an important aspect of processing natural language and of developing many text processing applications. Text segmentation can occur on different levels: low-level segmentation is carried out at the initial stage of text processing while high-level segmentation, being more linguistically motivated, could be defined as its main focus (Mikheev, 2004). Tokenization and sentence splitting are low-level segmentation tasks which are usually carried out by scripts consisting of regular expressions written in Perl, Flex or Python. Intra-sentential segmentation, like syntactic chunking or named entities recognition, represents an example of high-level text segmentation as much as inter-sentential segmentation which involves grouping sentences and paragraphs into discourse topics.

For the purpose of this thesis, only low-level text segmentation tasks will be discussed. First, I cover the topic of sentence boundary disambiguation which allows for the segmentation of text into sentences. Then, I address the one of tokenization.

Sentence boundary disambiguation

Sentence splitting, also known as sentence boundary disambiguation (SBD), consists of segmenting text into sentences by making use of punctuation marks, like periods, question marks and exclamation marks, that usually signal a sentence boundary. An accurate analysis of the local context around periods and other punctuation is hence of crucial importance for splitting text into sentences. Structural markup such as sentence tags can also be exploited, if present, to achieve this goal.
The main source of ambiguity when splitting a text into sentences is certainly constituted by abbreviations.

> *Appareils de vente automatiques […] tels que distributeurs automatiques de timbres-poste, cigarettes, chocolat, comestibles, etc.*
> *Voici M. Wilson et M. Rosener de Rockman Aviation.*
> *Art. 1 Champ d'application*

Since a period may signal the end of a sentence but also participate in the construction of an abbreviation, establishing whether the word preceding a period is just a shortened version, or not, will help in determining the sentence boundary. Another way to solve this

type of ambiguity would be looking at the word following the period in question, but if this word is capitalised, one could be faced again with a disambiguation problem: is that capitalised word a proper noun that goes along with the preceding abbreviation or does it constitute the beginning of a new sentence? Other sources of ambiguity for sentence boundary detection are typographical errors, such as missing whitespaces and quoted sentences appearing within main sentences, although the lowercase word following any citation or direct speech should be a clear signal that the sentence is not finished yet.

> *Commande jamais reçu.payement non remboursé.*
> *C'est, prétend-il, une mer sujette à d'affreux ouragans, semée d'îles inhospitalières, et « qui n'offre rien de bon » ni dans ses profondeurs, ni à sa surface.*

An erroneous sentence break may cause major categorization errors during POS tagging, while an accurate sentence splitting will benefit the linguistic annotation. But how is this achieved? Mikheev (2004) describes two main approaches to sentence boundary disambiguation as rule based and statistical. Rule-based disambiguators consist of a series of rules in the form of regular expressions, known also as regex, which are manually written and usually supplemented by external lists of words, the most common being abbreviations, but sometimes even proper names, common nouns and so on. Here is an example of a regular expression as it can be found in the Perl file of a rule disambiguator:

@regex_boundary = ('\.(?=[\s\t])', '\.$')

> *This expression tells the disambiguator that the period "\." has to be considered as a sentence boundary, and therefore can be segmented, only if it is followed by a space "\s", a tab "\t", or the end of a line "$".*

Nonetheless, it is well-known that developing systems based on rules is time-consuming, besides, these are usually tailored to a specific corpus. Automatic disambiguators, instead, have the advantage of being retrainable for a new corpus, text typology or for other languages. They are usually divided into two main groups - supervised and unsupervised. The most recent systems often make use of machine learning techniques like decision-tree classifiers, maximum entropy models, or neural networks which are essentially the same algorithms employed for part-of-speech tagging. The reason is that they treat sentence boundary disambiguation as a classification problem and make use

of features such as capitalization, suffixes, or word classes (Mikheev, 2004). The only drawback is that they require already labelled text for supervised training. On the contrary, there are statistical systems that can be trained from unannotated raw texts: the concept underlying their algorithms is that only a small portion of the periods found in a text are ambiguous, therefore regularities can be learned from unambiguous usages. Although the core of these statistical systems is language independent, it can be augmented with language specific add-ons to achieve a higher degree of accuracy (Mikheev, 2004).

Tokenization

If we are to set forth a down-to-earth definition of an electronic text, we would say that it is a sequence of content characters, such as letters, numbers, punctuation marks, symbols, and similar entities, that also contains control and typesetting characters like whitespaces, new lines, carriage returns, et cetera (Mikheev, 2004). Tokenizing, better known as tokenization, is the process for which a sentence or a text is segmented into linguistics units which are precisely the words, numbers and all other content characters mentioned in the above definition (Mikheev, 2004). These segmented units are called tokens.

Tokenization is a relatively easy process for alphabetic languages such as French since words are usually separated by whitespaces. However, this is not the case for ideographic languages, for example Chinese, where characters acting as word boundaries are absent. As a matter of fact, a standard tokenizer for alphabetic languages would achieve a reasonable degree of accuracy simply by replacing whitespaces between words with new lines and by cutting off punctuation marks from both ends of a word while adding blanks between them. As mentioned at the beginning of this section, such operations are performed through a script containing a series of regular expressions. However, designing a series of rules to overcome all tokenization challenges and to cover all possible exceptions found in texts of diverse typology is not a straightforward task even for the simplest writing system. Hence, more advanced techniques are often implemented to complement standard tokenizers.

Since not all tokenization modules are crafted in the same way, one must foresee in some cases a means for tokenizing the input text or corpus according to certain requirements before feeding it into the system.

Major tokenization challenges for French are presented below. While most of them concern the word segmentation issue, others such as period disambiguation and missing whitespaces also concern sentence boundary detection. Considerations about these two problems have been given in the subsection above, thus they will not be discussed any further.

CLITICS

French clitics include mostly pronouns and adverbs. Here follow some examples where clitics have been underlined to ease their detection:

“Permettez-vous ?”
"D'autres n'ont même pas pu arriver jusque-là."
"L'auteur de ce livre a-t-il pensé au scandale qu'il allait générer ?"

Clitics depend phonologically on other words and *clitic-clusters* (Seuren, 2009) can be easily mistaken for a single token. The segmentation of clitics in French is rather simple because personal pronouns attached after the verb can be detected by suffix matching, as long as exceptions such as “*rendez-vous*” are taken into account.

CONTRACTIONS/AMALGAMS

Another example of two or more tokens that could be mistaken for one are contractions and amalgams, known in French as *amalgames*. In this category are included both phenomena resulting from a contraction between:

- a determiner, relative pronoun, negative particle, conjunctions, etc. and a word starting with a vowel [contractions]; in this case the determiner’s vowel or the *silent e* of the relative pronoun/particles falls, and it is replaced by an apostrophe:

  La Commission est forte lorsqu'elle agit collégialement.
  La tablette n'est pas un modèle de réactivité.
  Nous l'avons posée sur le frigo pour y inscrire les aliments qu'il faut acheter.

- French prepositions *à/ de* and the definite articles *le*/*les* [amalgams]:
  Art. VI Dispositions d’application du présent Accord
  Nous l’avons reçu avec des pièces manquantes.

- French prepositions *à*/*de* and the different forms of *lequel* [amalgams]:

Découvrez les événements auxquels le premier ministre participe.

The first type, that is contraction, can be handled with the same technique used for clitics. On the contrary, the second type, which is usually referred to as amalgam, can be either

- left as it is, a part of speech is assigned to it, and the two words forming the amalgam are given in their canonical form in the column where lemmas should appear; or
- the amalgam is split, leaving two separate tokens to which is attributed a POS label.

WORD-INTERNAL PUNCTUATION

In written texts, expressions referring to one or more objects at the same time are quite common, as shown in the example below:

"Indiquer les nom et prénom et l'adresse complète de la (ou des) personne(s) ou société(s) …"

If a tokenizer is instructed to cut off any parenthesis before and after a word, in the case presented above, it will produce two incorrect tokens, that is *personne(s* and *société(s*. To avoid this type of error, the tokenizer should be provided with additional instructions allowing to check for a complementary symbol inside the word and to avoid splitting in this particular case.

Another punctuation mark which requires particular attention is the hyphen: French has a large number of words containing internal hyphens, such as *non-fumeur*, *entre-temps*, *abat-jour*, etc. These words do not need to be split because they constitute a single token, hence, when designing a tokenizer for French, strategies to deal with these cases should be carefully developed.

MULTIWORDS

So far, it has been assumed that tokens do not contain whitespaces but rather correspond to single words, numbers, punctuation marks, etc., or to two or more words linked by a hyphen or an apostrophe. This assumption was an oversimplification because French is a language rich in multiword expressions, also called MWE, which are formed by two or more words separated by whitespaces. Examples of MWE are

- coordinating conjunction such as "en effet", "par conséquent", "de plus";
- subordinating conjunctions like “avant que”, “de façon que”;
- complex preposition such as “à cause de”, “hors de”;
- idioms like “cul de sac”, " boîte noire" “ table ronde”;
- terms like "trou noir";
- named entities such as “Tour Eiffel”, “L'Oréal Paris”;
- cardinal numbers like "1 000" and "23 000 605";
- numerical expression, such as dates, and time expressions like “il y a”, "jusqu’à".

The main issue, as pointed out by Constant et al. (2017), is that “MWEs consist of several words (in the conventionally understood sense) but behave as single words to some extent”. These lexical units are very hard to predict since there is not a standard pattern that could be used to identify them. Hence, the exploitation of lexical resources is one of the main solutions for their recognition (Savary, A., Cordeiro, S. R., & Ramisch, C., 2019). Multiword expressions handling is crucial for NLP applications: although literature on computational linguistics seems to focus more on MWE in relation to parsing (Green et al., 2011; Candito & Constant, 2014; Simkó et al., 2017) where dependency relationships between constituent groups come into play, researchers (Variš & Klyueva, 2018) have also been trying to find a solution to improve the identification of MWE also with POS tagging.

1.2.2 Lemmatisation

In broader terms, lemmatisation is the process of reducing a word form to a more generalised representation, called lemma or base form. For example, a lemmatiser would attribute to the linguistic units or tokens in the sentence “*J'ai acheté cet article pour mes amis*” (= I bought this item for my friends) the following lemmas:

J'	→	*je*		*article*	→	*article*
ai	→	*avoir*		*pour*	→	*pour*
acheté	→	*acheter*		*mes*	→	*mon*
cet	→	*ce*		*amis*	→	*ami*

Fitschen and Gupta (2008) distinguish two variants of automatic lemmatisation – one that requires lexical information and one that does not. The lexicon-based lemmatisation

variant can be further subdivided according to the approach used: lexical information can either be extracted from an exhaustive list of potential base forms or be generated according to an existing paradigm, built on theoretical assumptions and conventions. Conversely, the lemmatisation variant which does not require a lexicon is called stemming. The lemma, or *stem*, is obtained by truncating the original word according to a relatively arbitrary set of rules. Stemming does not rely on morphological analysis, and it could cause the removal of both inflectional and derivational endings, thus the resulting lemma may not be linguistically motivated.

In this thesis, I am concerned with the linguistically founded lemma, that is with the base (or canonical) form of a word stripped from its inflectional affixes. In other words, the form of a word as it is found on a dictionary.

Although some part-of-speech taggers do not provide for the lemmatisation tasks, its output is very useful in some concrete applications of natural language processing such as corpus query.

1.3 POS tagging approaches

Different sequence labelling models for the automatic annotation of texts with parts of speech have been conceived over the years (Brill, 1992; Ratnaparkhi, 1996; Brants, 2000; Toutanova et al., 2003; Müller et al., 2013; Gui et al., 2017). Although these systems could be categorised in several ways, the general distinction made here is between the linguistic approach and the automatic data-driven approach (Voutilainen, 1999). This binary division might suggest that these systems are methodologically "pure". In fact, they are not, because data-driven approaches presuppose a certain extent of linguistic knowledge whereas linguistic approaches may use not only linguistic and heuristic rules to resolve disambiguation problems but also automatically induced rules.

Linguistic approach

The earliest part-of-speech taggers were based on hand-written disambiguation rules and date back to the late 1950s-early 1960s. These rules were made by expert grammarians and were based on generalisations about the language as well as on observations of text samples, descriptive grammars, and dictionaries (Voutilainen, 1999). These systems used a small lexicon containing all possible analysis to some words of the input text, while heuristic rules were used for all those words that were not represented

in the lexicon. Heuristic rules "relied on affix-like letter sequences at word-boundaries, capitalization and other graphemic clues about word category" (Voutilainen, 1999, p. 10). Words that were not analysed by either the lexicon or the heuristic rules were attributed several parts-of-speech as alternatives. The latter were subsequently eliminated by reductionist linguistic rules on the basis of the local context. For example, the following rule is about the wordform "*A*" that is ambiguous in French because it can be either the abbreviation of "*ampere*", or the preposition "*à*" at the beginning of a sentence, or the verb "*avoir*" in a sentence like "*A-t-il emporté la caméra avec lui ?*" (= Did he take the camera with him?).

"<A>" REMOVE (ABR)
*(NOT *-1 POS)*
*(*1 (CLITIC) OR (PRON) OR (ADJ) OR (ADV))*

*The abbreviation reading of "A" is discarded if it is the first word in a sentence (the term "POS" represents the set containing all parts of speech tags) and if the following word (*1) is a clitic (-t-il), a pronoun (e.g., tout), an adjective (e.g., plus) or an adverb (e.g., bientôt).*

After this process, if some words were still associated with more than one tag, a human posteditor would have corrected the output.

Data-driven approach

The main feature of the data-driven approach is the use of already annotated data to train the language model. These systems extract information regarding tagsets, frequencies of the word-tag pairs, sets of rules and so on, from pre-annotated corpora during a training phase; subsequently, they make use of these statistically extracted information to annotate raw data.

TRANSFORMATION-BASED

The so-called transformation-based approach, which is used by the Brill tagger, rely on a Transformation-Based Error-Driven Learning (Brill, 1995). This approach is at the frontier between different type of taggers: it is similar to a rule-based tagger, but its rules are automatically induced rather than being written in advance; and it is data-driven but the input text does not have to be already annotated.

In general, the process works as follows: first, raw text is passed through an initial-state annotator which assigns part-of-speech tags to the input words, this is frequently achieved by means of a stochastic method. Then, a manually annotated corpus used as a reference is submitted to the tagger which learns an ordered list of transformations rules, or correction rules, by comparing the two sets of data. Finally, these rules are applied to the initially tagged text to generate the final output.
Let assume, for example, that the initial-state annotator has erroneously tagged a noun in a nominal phrase as a verb:

Ordonnance/VERB du/ADP+DET DETEC/PROPN

In the corpus submitted to the tagger there are several noun phrases so that the system has learnt that it is possible to have the following sequence of tag "NOUN ADP+DET PROPN". The correction rule that the system created when confronting the two sets looks like this:

Change the tag VERB to NOUN if:
the following tag is ADP+DET
the tag following this word by two is PROPN

This process of applying transformations rules to the initially tagged text is usually reiterated a few times in order to improve the tagging performance. Moreover, every iteration allows the taggers to enhance an already learnt rule, thus improving even more the performance of the model.

HMMs

The traditional algorithm for sequence modelling is the Hidden Markov Model (HMM). Many of the key concepts introduced with this algorithm have also been employed in modern models. An HMM is a probabilistic sequence model that works as follows: given a sequence of words, the HMM "computes a probability distribution over possible sequences of labels and chooses the best label sequence" (Jurafsky & Martin, 2020, Chapter 8, p.8).
This model is based on augmenting the Markov chain, a stochastic process useful to compute the probability for a sequence of observable events. As Jurafsky & Martin clarify (2020, p.8) the Markov chain "tells us something about the probabilities of sequences of random variables, states, each of which can take on values from some set" and assumes

that to predict the future in a sequence of random variables, only the current state matters, while previous states have no impact on the prediction. In addition to this assumption, known as the Markov assumption, a first-order hidden Markov model instantiates the "output independence" assumption for which "the probability of an output observation depends only on the state that produced the observation and not on any other states or any other observations" (Jurafsky & Martin, 2020, Chapter 8, p.9).

The events, or variables, for which we want to compute the probability are the part-of-speech tags, but when we process a text, we deal with sequences of words (observations) and we cannot determine the exact sequence of states through which the model passes to generate those words' sequences (Jurafsky & Martin, 2020, Appendix A, p.11). The sequences of states are hidden and cannot be observed, that is why we refer to this model through the adjective "hidden". (Jurafsky & Martin, 2020, Appendix A, p.2). A hidden Markov Model allows us to work with both observed events (such as the words we see in the text provided as an input) and hidden events (such as the part-of-speech tags).

The task of generating the hidden variables sequence (tags) that corresponds to the sequence of observed events (words) is called decoding. The HMM uses the Viterbi algorithm for decoding, that is a dynamic programming algorithm for obtaining the maximum a posteriori probability estimate of the most likely sequence of hidden states. The transition and emission probabilities are calculated by the maximum likelihood estimation (MLE) which derives from the analysis of the pre-tagged training corpora.

One of the weaknesses of HMMs comes from the fact that the Viterbi algorithm requires a lot of memory and computation time since it computes a probability for each tag at each time step to determine the sequence of words that are associated to the most probable sequence of tags. Moreover, it is quite complex to incorporate arbitrary features to deal with unknown words in a generative model.

CRFs

A Conditional Random Fields (CRFs) is a discriminative probabilistic graphical model based on log-linear models (Sutton & McCallum, 2010). The version commonly used for language processing is the linear chain CRF.

Given an input sequence of words regarded as a whole, this model assigns a probability to an entire sequence of tags, out of all possible sequences. The CRF has a function which maps the entire input and output sequences to a feature vector. These global features are

afterwards decomposed in a sum of local features for each position in the output sequence of tags. The local features make use of the current and previous output tag to produce a global probability. Hence, contrary to the HMM, the CRF does not compute the probability of each single tag at each time step but rather estimates a log-linear function over the set of features previously mapped, which are subsequently aggregated and normalized to produce a global probability of the entire sequence of tags (Jurafsky & Martin, 2020, Chapter 8).

The advantage of the CRF is that it is easier to incorporate a lot of features in this model while the high computational complexity of the training stage of the algorithm is the major drawback.

NEURAL NETWORKS

Artificial neural networks consist of an assembly of simple processing elements, called units, which are highly interconnected by directed weighted links. Associated with each unit is an activation value that is propagated through these interunit connections (Schmid, 1994b). The processing ability of the network is stored in the interunit connection weights which are obtained by learning from a set of training patterns.

In the Feedforward Networks, also known as Multi-Layer Perceptrons (MLPs), the processing units are arranged vertically in several layers and connections exist only between units in adjacent layers. The bottom layer is called input layer because the activations of the units in this layer represent the input of the network. The top layer is instead the output layer. Any intermediate layer is called hidden layer. Schmid (1994b, p.173) argue that during the processing in an MLP-network, activations are propagated from input units through hidden units to output units. At each unit, the weighted input activations are summed up and a bias parameter is added. The resulting network input is then passed through a logistic function in order to restrict the value range of the resulting activation to the interval [0,1] (Schmid, 1994b). The network learns by adapting the weights of the connections between units, until the correct output is produced (Schmid, 1994b).

The part-of-speech tagging of a word with Net-tagger (Schmid, 1994b, p. 174), which consists of an MLP-network and a lexicon, is carried out as follows: firstly, the tag probabilities of the current word and its neighbours (the preceding and the following words) deriving from the training data are copied into the input units; activations are

then propagated through the network to the output units; finally, the tag corresponding to the output unit which has the highest activation is attached to the current word. If there is an output layer with an activation that is close to the highest one, the tag corresponding to the second strongest activation may be given as an alternative output (Schmid, 1994b). What differentiates the various types of neural networks is how the information passes through the network (Schmidt, 2019). While MLPs, pass information without cycles, the Recurrent Neural Network (RNN) has cycles and transmits information back into itself. This enables RNNs to extend the functionality of the MLPs to also take into account previous inputs and not only the current input (Schmidt, 2019). However, traditional RNNs face the vanishing gradient problem that can occur during training. A long short-term memory (LSTM) networks is an architecture that has been developed to deal with this problem, although it may encounter the exploding gradient problem.

Conclusion

In this chapter, that type of linguistic annotation allowing to input morphosyntactic information into a corpus, that is part-of-speech tagging, has been defined. Processes preliminary to POS tagging, such as text segmentation into sentences and tokenization, have been introduced along with lemmatization, another process which is sometimes integrated into tagging systems to make the linguistic information more complete. Finally, some of the main computational approaches used to perform part-of-speech tagging have been described: they have been classified into linguistic-based and data-driven methods, the latter including HMMs, CRFs and neural networks. All this information is useful to understand the purpose of POS taggers, which kind of approach they use, and what kind of processes they are expected to operate. The latter information is particularly relevant for the present evaluation.

We now move onto the topic of evaluation to explore the most common methods and measures applied to language processing systems in general, and part-of-speech taggers in particular.

2. Language processing systems evaluation

The evaluation of language processing systems as a mean to foster the development of research and technology in the field of language engineering became prominent towards the end of the 1980s in America and in the mid-1990s in Europe with the organisation of the first series of evaluation campaigns (Paroubek et al., 2007). The "evaluation paradigm" (Adda et al., 1998), although it had been initially and mainly applied in the United States, was soon adopted in Europe. This evaluation paradigm comprises two phases: the first phase consists of the preparation of the data that are then exploited to create the systems to process them; the second phase consists of a series of tests that allow for the comparison of the systems on similar data. At last, the results of these tests and the discussion they generate become the foundation of the evaluation (Adda et al., 1998).

From that point onward, different methodologies for the evaluation of language processing systems have seen the light. As Paroubek (2007) points out, among the general characterizations of evaluation found in the literature, the main characteristics of evaluation methodologies are:

1. *Black box* versus *white box* evaluation – the first one presupposes that only the global function of a system is accessible whereas the second one presupposes that all its subfunctions are also accessible for examination.
2. *Objective* versus *subjective* evaluation – the first type implies that measurements are performed directly on the data produced by the process being tested whereas the second one implies that the measurements are based on the perception that individuals have of this process under test.
3. *Qualitative* versus *quantitative* evaluation – the first one presupposes that the result is a label which describes the behaviour of a system whereas the second one presupposes that the result is the value of the measurement of a specific variable.
4. *Technology, or system-oriented,* versus *user-oriented* evaluation – even if the distinction between these two types is less clear (see Paroubek et al., 2007), the first one refers to the measurement of the performance of a

system on a generic task while the second one refers the way real users utilize the system.

The choice of an evaluation's method is certainly influenced by the system or system's components subjected to the evaluation, the software life cycle and the major stakeholders involved, but Hirschman & Mani (2004) argue that the style of an evaluation also depends on the inputs and outputs of the system in question. In accordance with this claim, they organize natural language processing systems in three classes: (1) analysis systems, for example POS tagging and parsing to name but a few; (2) systems that produce a language output, such as translation and generation systems; and (3) interactive systems, "where user and system exchange information through a multi-turn interaction to achieve a goal" (Hirschman & Mani, 2004, p.416).
A*nalysis systems* are defined by Hirschman & Mani (2004, p.415) as systems which "accept a language input and produce an abstract representation or classification of that input". One of the most common methods to evaluate these language technology systems or components is by means of a benchmark, known also as gold-standard: the output of the system is compared against its gold-standard and a performance score is assigned (*comparative evaluation*). Another useful mean for the evaluation of a language system are feature-based metrics: the most significant example is the method established by the European community, namely the EAGLES 7-step recipe (EAGLES, 1999).

The rest of this chapter is structured as follows: gold-standard-based measures for part-of-speech tagging are presented in detail in 2.1, feature-based metrics for the evaluation of language products are introduced in section 2.2 while a brief account of state-of-the-art evaluation for POS tagging is given in 2.3.

2.1 Gold-standard-based measures

In the context of POS tagging, a gold-standard is essentially a version of the sentence, text, or corpus to be tested that has already been annotated and against which the output of a system is evaluated. The gold-standard usually undergoes an automatic or semi-automatic annotation stage followed by a revision stage for consistency checking operated by one or more human annotators. While the automatic or semi-automatic annotation may be foregone, the human revision phase is essential to ensure consistency

in the annotation which is the most important factor in determining the quality of an annotated resource (Heike Zinsmeister et al., 2008).
As outlined by Hirschman & Mani (2004, p.417), gold-standard-based (or comparative) evaluation measures usually consists of the following stages:

- Definition of the evaluation task and of a gold-standard format. The latter requires the development of annotation guidelines, ideally a tool to support the annotation process, and the validation of that process through the calculation of the inter-annotator agreement (Ron, 2017). The inter-annotator agreement's score allows to assess the reliability of the annotation, as a precondition for ensuring its correctness. Among the various measures of inter-annotator agreement are Cohen's κ, used when there are only two annotators assigning each token with a label, and Fleiss's κ, that estimates the proportion of labels on which two or more annotators agree (Ron, 2017).

- Preparation of annotated training, development, and test corpora. To avoid misleading results, the corpora used during the training, development and test phases must contain different data.

- Evaluation of the system by comparing the processed corpus against its gold-standard which results in the attribution of a score.

In the case of part-of-speech tagging, when measuring the performance of a system directly on the data produced, typical metrics as mentioned in the literature are accuracy, precision/recall, and error rate (Hirschman & Mani, 2004). The most intuitive and most used metric is certainly the accuracy which is defined as "the ratio of the number of word forms correctly tagged over the total number of word forms tagged" (Paroubek, 2007, p.110). The value of precision and recall are respectively "the ratio of the number of correct tags over the number of tags assigned by the system" and "the ratio of the number of correct tags over the number of tags assigned in the reference" (Paroubek, 2007, p.112). Finally, the error rate is nothing more than the complementary value of the accuracy score, thus if the accuracy of a POS tagging system is 89.5%, the error rate will be 10.5%.

Evaluation measures based on gold-standards are an example of a quantitative evaluation since the resulting score is attributed from the measurement of a particular variable (Paroubek, 2007), but also of an objective evaluation since measurements are performed on the data produced by the process being tested, that is the part-of-speech annotation in this case. They are also common measures of black-box evaluations.

2.2 Feature-based metrics: EAGLES

ISO/IEC SQuaRE (System and Software Quality Requirements and Evaluation)[2], also known as ISO/IEC 25000, is a series of International Standard for the evaluation of software product quality devised by The Expert Advisory Group on Language Engineering Standards (EAGLES). Among the five divisions that are part of this series, the *ISO/IEC 2501n* is the one that "present detailed quality models for computer systems and software products, quality in use, and data" (ISO/IEC 25010, 2011, p. vi). The Quality Model Division is further subdivided into two standards:

- ISO/IEC 25010 - System and software quality models: Describes the model, consisting of characteristics and sub-characteristics, for software product quality, and software quality in use.

- ISO/IEC 25012 - Data Quality model: defines a general data quality model for data retained in a structured format within a computer system. It focuses on the quality of the data as part of a computer system and defines quality characteristics for target data used by humans and systems.

Along with these standards EAGLES has also devised some guidelines for the implementation of a solid evaluation of systems or system's modules that are based on language technologies, namely the *EAGLES 7-step recipe* (EAGLES, 1999). These seven steps consist in

1. Defining the scope of the evaluation and the stakeholders it addresses.

2. Elaborating a task model, namely defining which tasks are going to be investigated and which system's features can accomplish those tasks.

[2] https://iso25000.com/index.php/en/iso-25000-standards [Retrieved June 23rd, 2021]

3. Defining the top-level quality characteristics, that is identifying which features need to be evaluated and what is their relevance.

4. Producing detailed requirements for the system under evaluation. It could be necessary to identify some quality sub-characteristics if the features defined in the 2nd and 3rd step are not directly measurable until this becomes possible.

5. Devising the metric to be applied to the system for the requirements produced, including methods, for example benchmarking, feature inspection or scenario testing, and measures such as measurements units, true/false scales, rating scales, etc.

6. Designing the execution of the evaluation. This step entails the development of an evaluation protocol, test material, and the identification of the participants, among others.

7. Executing the evaluation and summarizing the results.

To define the characteristics and sub-characteristics at point 3 and 4, one must refer to the International Standards mentioned above. This is how the guidelines and the standards defined by EAGLES merge to form the quality evaluation framework based on features. Besides, if we want to go back to the categorisation of evaluations given at the beginning of chapter 2 and consider the international standard ISO/IEC 25010:2011 (E) which includes the models of *quality in use* and *product quality*, we could say that the former is a user-oriented type of evaluation, while the latter is system-oriented.

2.3 State-of-the-art evaluation for POS tagging

Over the years, various evaluations for French have been carried out following the methodologies seen above. One of the first examples entails the implementation of the evaluation paradigm (Adda et al., 1998) during the GRACE campaign on morphosyntactic taggers. Other examples are comparative evaluations: to be defined as such an evaluation requires “standard and common ground linguistic resources for both training and testing tasks" (Zeroual & Lakhouaja, 2019, p.2) and that “a part of the corpus is excluded from the training data to provide an unseen test set” (Allauzen & Bonneau-Maynard, 2008, p.1).

Confident comparative evaluations on POS tagging are provided, for example, by Allauzen & Bonneau-Maynard (2008) for French, and for other languages by Horsmann et al. (2015) and Zeroual & Lakhouaja (2019). Allauzen & Bonneau-Maynard (2008) present a comparison of three statistical POS taggers for French which have been trained and evaluated in the same conditions. The linguistic resource used is the French MULTITAG (Paroubek, 2000) corpus, a large resource of 1 million words with a rich tagset that contains also inflectional features, such as gender, number, etc. Horsmann et al. (2015) present instead a comparison of 22 POS taggers models for English and German given by 9 different implementations. The approach however is slightly different since they use several corpora and excludes from the testing phase the ones with which the tagging models have been trained. The latter comparison is on POS tagging for Arabic provided by Zeroual & Lakhouaja (2019). Two corpora, classic and modern Arabic, are used and are split in a 90% of words for training and 10% for testing.

Conclusion

In this chapter, the evaluation of language processing systems and its different categorisations have been presented along with two evaluation methods which can be applied to part-of-speech tagging based respectively on gold-standards and on features.
In the next chapter, the experimental setup of the present evaluation is given: the two methods, based on gold-standards and on features, are combined in an evaluation framework which follows the EAGLES guidelines (EAGLES, 1999) and applies the ISO/IEC 25010:2011 Quality in Use Model to assess the quality of some properties of three part-of-speech taggers for French. The metrics and methods at the core of this evaluation are discussed in detail and the corpora used for the experiments are presented as well as the POS taggers to be evaluated. More examples of past comparative evaluations on French POS tagging are also given with respect to the part-of-speech taggers involved in the present evaluation along with an account of the metrics used, and the results obtained.

3. Experimental Setup and methodology

The goal of the present study is to evaluate the performance and use of three part-of-speech taggers for French, namely MElt, TreeTagger, and UDPipe 2.0. The focus is distributed on three sub-questions: how is the tagging performance affected by a specific text typology? Which system's interface is more user-friendly? Are the taggers provided with embedded modules that are able to deal with various file formats and cover all stages of text pre-processing and processing? To find an answer to these questions two series of experiments are carried out in which the taggers are tested without training or tuning, therefore their integrated French language model is used:

First phase of experiments

> Taggers are tested on four corpora (Spoken, Literature, Review, and Law) given in an XML format. If the XML corpora cannot be processed, their equivalent plain text versions are used. The taggers operate the segmentation of sentences into tokens, annotate the resulting tokens with POS tags and provide a lemma for each token.

Second phase of experiments

> The taggers are run on the Evaluation corpus and set to carry out only the annotation in parts of speech. Their outputs are then compared against a reference corpus, namely the Gold-standard.

In this chapter, we look at the methods and metrics at the core of this evaluation (3.1), the test corpora used for the two series of experiments (3.2) and the annotation of the Gold-standard used as a reference corpus in the second series (3.3). Finally, the POS taggers to be evaluated are presented and an account of their annotation strategy as well as their internal tagset is given (3.4).

3.1 Quality Model

The Quality model developed for this evaluation on three part-of-speech tagging systems for French is based on elements from the ISO/IEC 25010:2011 (E) (SQuaRE) series of Standards (International Organization for Standardization / International Electrotechnical Commission, 2011). In the reminder of this section, the characteristics

of the Quality in Use model providing the framework for the present evaluation are illustrated, following the EAGLES guidelines (EAGLES, 1996) (see section 2.2), along with the quality requirements that each system must fulfil with respect to the specified context of use.

1. The scope of the evaluation is to identify which POS tagging system for French among TreeTagger, MElt and UDPipe 2.0:
    a. has the easiest implementation or the most user-friendly interface, with respect to the defined stakeholders, namely students, translators, and researchers without a solid background in computer science.
    b. is able to process corpora stored in an XML format which contain structural XML tags in addition to corpora stored in a plain text format.
    c. embeds all the modules normally required to perform the pre-processing and processing of corpora (tokenization, POS tagging and lemmatization).
    d. produces the best annotation for a corpus of a specific text typology (speech transcripts, literature, product reviews and legal texts)

2. To fulfil the scope, three main tasks have been identified:
    - Taggers are downloaded and setup (if necessary) on a Windows Operating System.
    - The POS taggers are run on the four corpora (Spoken, Literature, Review, and Law) presented in section 3.2.1, which are stored in an XML format and contain structural tags. If XML corpora cannot be processed, their plain text versions are used. The taggers operate the segmentation of sentences into tokens and the annotation of the resulting tokens with POS tags and lemmas.
    - The taggers are run on the Evaluation corpus (section 3.2.2) which is arranged in one-token-per-line and stored in a plain text format. Taggers annotate the Evaluation corpus only with POS tags.

3. Given the task model outlined in point 2, the three top-level characteristics of the "Quality in use" model that will be investigated are *efficiency*, *effectiveness*, and *context coverage*. Context coverage is a broader characteristic which comprises *context completeness* and *flexibility* as sub-characteristics. However, given that this study does not aim at evaluating the systems in contexts beyond those initially specified, only the *context completeness* sub-property is investigated.

4. The *efficiency* of a system is given by the "resources expended in relation to the accuracy and completeness with which users achieve goals" (ISO/IEC, 2011, p.8). A part-of-speech tagger would be considered efficient in the current context of use if it includes all the following modules: tokenization (including correct handling of MWEs) and lemmatization. In addition, given that electronic texts are nowadays stored in a variety of formats, the most used ones being XML and plain text, a system will be considered efficient if it can handle both these text formats.
   The *effectiveness* of a system is defined as the "accuracy and completeness with which users achieve specified goals" (ISO/IEC, 2011, p.8). A part-of-speech tagger would be considered efficient in the current context, if it assigns the correct POS tag to all data submitted to it.
   Context completeness concerns the "degree to which a product or system can be used with effectiveness, efficiency, […] in all the specified contexts of use" (ISO/IEC, 2011, p.9). Given that the stakeholders addressed in this evaluation are students, translators, and other researchers with limited competence in programming, a part-of-speech tagger would satisfy the context completeness sub-characteristic if it were provided with a graphical user interface or a web-based interface through which users can confidently carry out the POS tagging task.

   Table 6 provides a summary of the two characteristics and one sub-characteristic of the Quality Model adopted and the respective system's property which is investigated:

CHARACTERISTIC OR SUB-CHARACTERISTIC	PROPERTY OF THE SYSTEM
Efficiency	1 – Presence or absence of pre-processing and processing modules such as tokenization (including MWE handling) and lemmatization 2 – Ability or inability to handle file formats such as XML and plain text
Effectiveness	Part-of-speech tagging performance
Context coverage → Context completeness	Presence or absence of a graphical interface or a web-based interface

Table 6. Quality model characteristics and properties of the taggers to be evaluated

5. The *efficiency* of the systems is evaluated by means of a feature inspection (section 4.1.2) that is carried out when testing different corpora. This aims to ascertain the presence or absence of the modules usually embedded within the architecture of a POS tagger (tokenization, lemmatisation, and tagging). Furthermore, it allows to determine if the modules can handle both XML and plain text files. The metric used is a binary Yes/No answer with respect to the two questions "is the module present?" and "are the modules able to deal with both XML and plain text files?" These answers are given by the author of the present evaluation.

   A quantitative evaluation of the taggers' performance on different text typology (section 4.2.1) allows to assess the *effectiveness* of the systems considered. The metric used is the accuracy which is measured as the ratio of the number of tokens correctly tagged over the total number of tokens tagged (Table 12, section 3.2.2).

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

   Finally, *context completeness* is rendered through an inspection (section 4.1.1) of the interfaces available by which the user can interact with the systems to achieve the goal of annotating the data at its disposal. The metric used is a binary Yes/No answer with respect to the question "does the system provide a graphic or a web-based interface?". The answer is given by the author of the present evaluation.

   Table 7 provides a summary of the characteristics of the Quality Model adopted and the metrics used to evaluate them:

CHARACTERISTIC OR SUB-CHARACTERISTIC	METRIC
Efficiency	1 – Yes/No answer 2 – Yes/No answer
Effectiveness	Accuracy
Context coverage → Context completeness	Yes/No answer

Table 7. Quality model characteristics and metrics used for their evaluation

6. The present evaluation on three part-of-speech taggers for French is organised in two series of experiments. In the first series, the first two main tasks outlined in point 2 are performed: taggers are implemented (if necessary) and deployed on four corpora (3.2.1) to determine their ability to deal with different file formats and to give account of the modules embedded within their architecture (*efficiency*) as well as to test the degree of user-friendliness of their interface (*context completeness*). In the second series of experiments, the third main task is carried out: taggers are run on the Evaluation corpus (3.2.2) and their output is compared against the Gold-standard corpus (3.3). Various accuracy scores are then computed to assess the POS tagging performance of each system on the different text typologies available (*effectiveness*).

A summary of the framework at the core of the present evaluation is in given in Table 8:

EXPERIMENTS PHASE	CHARACTERISTIC OF THE QUALITY MODEL	TASKS	PROPERTY OF THE SYSTEM	METRIC	TYPE OF METRIC	TYPE OF EVALUATION
1st	Context coverage → Context completeness	Setup and deployment of the taggers	Presence or absence of a graphical interface or a web-based interface	Yes/No answer	Subjective	Qualitative
	Efficiency	The three taggers are tasked with the annotation in POS and lemmas of four XML or plain text corpora (Spoken, Literature, Review, and Law)	1 – Presence or absence of pre-processing and processing modules such as tokenization (including MWE handling) and lemmatization 2 – Ability or inability to handle file formats such as XML and plain text	1 – Yes/No answer 2 – Yes/No answer	Objective	Objective
2nd	Effectiveness	The three taggers are tasked with the annotation in POS of the Evaluation corpus. Outputs are compared against the Gold-standard.	Part-of-speech tagging performance	Accuracy	Objective	Quantitative

Table 8. Evaluation framework of part-of-speech tagging for French

3.2 Test corpora

The creation of annotated corpora has been a trending topic for the past three decades due to its manifold purposes and applications in several fields like linguistic and translations studies, language teaching and learning, language engineering, etc. Annotated corpora have been largely employed, for example, as a means to inform dictionaries and grammar books, to extract information, to raise students' awareness on language use as well as a medium for developing and improving natural language processing applications (Leech & Smith, 1999), such as part-of-speech taggers.

The corpora devised for the present thesis essentially serve two purposes: firstly, they help to understand how the systems considered behave and what are the requirements they must fulfil in order to be processed correctly; secondly, they allow to assess the systems' performance on different domains. The corpora used during the first series of experiments which allow to achieve the first purpose are described in 3.2.1. The construction of the Evaluation corpus which serves in the second series to fulfil the second purpose, that is a quantitative evaluation of the taggers' performance, is presented in 3.2.2.

3.2.1 Corpora (1st phase)

For the first round of experiments, four corpora have been created, one consisting of speech transcripts and three others consisting of written texts from diverse genre or styles: a few chapters from an 18th century fiction book stand for the literary writing style as opposed to commercial products reviews which render the informal style; finally, a small collection of legal texts represents instead the legal writing style.

These text typologies have been chosen because of their specificities which constitute a challenge for part-of-speech annotators. In the speech transcript corpus, for example, these challenges are the broken syntax, repetitions, broken words ending with hyphens, and the various labels used for the anonymisation of the speakers which usually contain a variety of characters, such as numbers, letters, and symbols. On the opposite side, there is the literary corpus that could potentially be the least problematic of all, if it wasn't that it dates to the late 1800s. Words which are no longer in use or less frequent may thus be found as well as linguistic variations and formal syntactic structures. Typical features of the review corpus are instead colloquialisms, figures of speech, emojis, typos and again broken syntax. Finally, the law corpus is the one that contains more technical terms in

comparison to the other three as well as named entities and text-specific features, such as numbered lists, abbreviations, and longer sentences.

Table 9 provides an overview of the number of words (approximately 5000) and tokens present in each corpus: these values are obtained through the uploading and compilation of the four corpora on the web-based program SketchEngine[3].

CORPUS	WORDCOUNT	TOKENS
Spoken	~ 6,091	~ 6,589
Written (Literature)	~ 5,286	~ 6,272
Written (Review)	~ 5,105	~ 5,774
Written (Law)	~ 5,297	~ 6,312

Table 9. Corpora 1st phase: words and tokens count.

Let us now take a closer look at the origins and content of these 4 corpora.

Spoken

The Spoken corpus (Figure 1) contains random transcripts issued from TCOF corpus[4] (ATILF, 2020), namely a broad collection of recordings of spoken French and their audio transcription (see Annex 1). The selected recordings concern spontaneous interactions between adults in diverse settings: a public context, as in the case of the general meeting of a pétanque club recorded in 2008, and a professional context, as for the interview of a professional rock climber. Each recording is distributed along with two files: (1) an XML files containing general information regarding the transcription, the recording, and the anonymised speakers; (2) a TRS file containing the actual transcript which is organised according to descriptive tags (XML elements with one or more attribute-value pairs), such as <turn>, <event>, <comment>, etc.

Literature

The Literature corpus (Figure 2) is composed of random chapters from the 1869's book *Vingt mille lieues sous les mers* by Jules Verne. This French novel is part of the opensource parallel corpus *ParCoGLiJe*[5] (Stosic & Miletic, 2019) (see Annex 1). The entire corpus can be downloaded from the ORTOLANG website: it contains the English and French versions

[3] https://www.sketchengine.eu/ [Retrieved March 28th, 2021]
[4] https://tcof.atilf.fr/index.php [Retrieved March 8th, 2021]
[5] https://www.ortolang.fr/market/corpora/stosic/ [Retrieved March 8th, 2021]

of several books which are distributed in an XML format annotated with a TEI-P5 standard.

Review

The Review corpus (Figure 3) contains randomly selected reviews from the French dataset of the *Multilingual Amazon Reviews Corpus*[6] (Keung et al., 2020). The latter is available for academic purposes to those users willing to open an account on Amazon Web Service (AWS) and is subjected to the Amazon.com Condition of Use[7]. Reviews were collected during the time span 2015-2019 and stored in JSON format with metadata.

Law

The Law corpus (Figure 4) contains a small collection of texts issued from the French version of *CHEU-lex*, a parallel and comparable trilingual corpus of Swiss and EU Legislation (see Annex 1). *CHEU-lex* gathers texts written in the time span 1972-2017, contains several levels of annotation (structural, part-of-speech, and grammatical) and is accessible through the Transius website[8].

As we do not know in advance if the selected taggers are able to process both XML and plain text files (reason for which this ability is going to be evaluated) two versions have been created for each one of the four corpora presented above (see Annex 2). In the following sub-sections, the creation and the structure of the XML and plain text corpora are detailed.

[6] https://github.com/awslabs/open-data-docs/tree/main/docs/amazon-reviews-ml [Retrieved June 3rd, 2021]
[7] Amazon.com - Conditions of Use [Retrieved July 30th, 2021]
[8] https://transius.unige.ch/en/research/cheu-lex/ [Retrieved July 30th, 2021]

XML versions (see Annex 2)

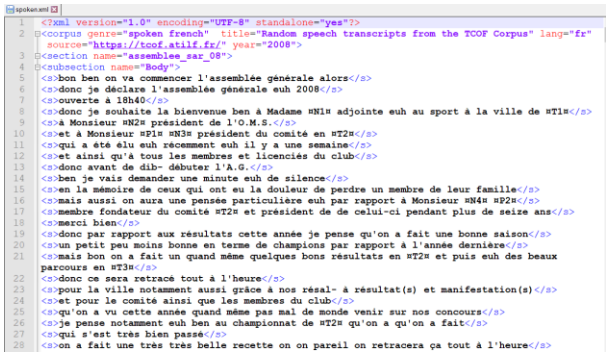


Figure 1. Spoken corpus XML

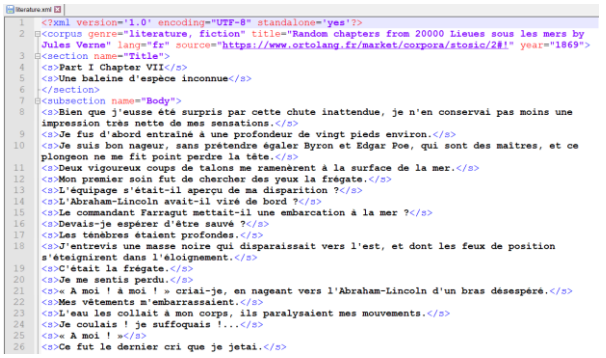


Figure 2. Literature corpus XML

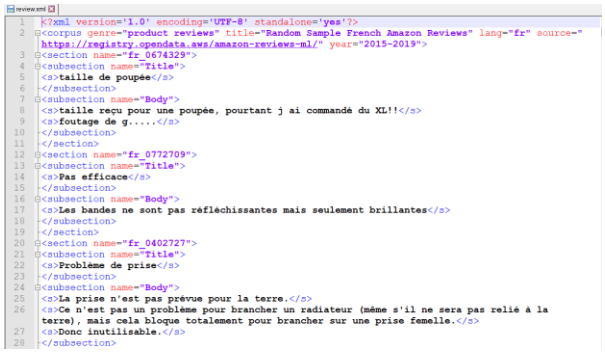


Figure 3. Review corpus XML

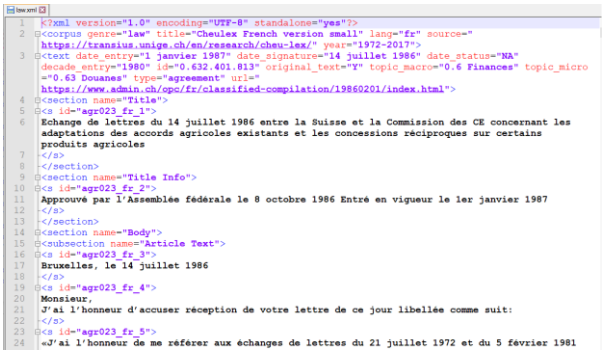


Figure 4. Law corpus XM

None of the original metadata and other structural XML tags has been preserved for the Spoken, Literature and Review corpora. On the contrary, texts have been manually annotated by the author of the present thesis through the software Notepad++[9] using regular expressions. Each one of these three corpus is arranged according to <section> and <subsection> tags with an attribute "name" which takes on different values. The value of attribute "name" of the element "section" can be

- the name of the transcription for Spoken,
- "Title" for Literature, and
- the ID number of the reviews for Review.

The value of the attribute "name" of the element "subsection" is either "Body" or "Title" (the latter is used only for the Review corpus).

In addition to these tags, these 3 corpora have also been annotated with sentence tags <s> </s> by means of different expedients: sentence tags have been manually added in the Spoken corpus using regular expressions on Notepad++ while the segmentation into

[9] https://notepad-plus-plus.org/ [Retrieved February 15th, 2021]

sentences of the Review and Literature corpora has been performed using Intertext Editor[10], a software that is conceived for aligning parallel texts, but which provides a very useful functionality for the segmentation of texts into sentences.

With regard to the Law corpus, the original XML annotation of the texts extracted from the French sub-corpus of CHEU-lex has been maintained. It includes contextual information and structural features such as *title*, *preamble*, *articles*, *annex*, to name but a few. The original sentence tags along with their attribute(*id*)-value pairs have also been preserved.

All XML corpora have been validated by means of an online XML validator[11] to make sure the documents are well-formed.

Finally, with respect to encoding standards, it should be mentioned that the original JSON file downloaded from the AWS platform from which the French reviews have been extracted, contained emojis and some accented letters written as UTF-16 surrogate pairs. The following python script has been used to convert the texts into the "latin-1" encoding, also known as "iso-8859-1", then into "UTF-8".

```
import unicodedata
import codecs
text = u"unicode text here"
modified = unicodedata.normalize(u'NFKD', text).encode('utf-16',
'surrogatepass').decode('iso-8859-1')
new_text = codecs.open('mynewfile.txt','w', 'iso-8859-1')
new_text.write(modified)
```

Despite this processing stage which successfully converted all emojis, accented letters were still represented by a combination of a letter and a separate accent mark, such as

` + a = à ; ´ + E = É ; ^+ o = ô.

[10] https://wanthalf.saga.cz/intertext [Retrieved June 1st, 2021]
[11] https://www.xmlvalidation.com/ [Retrieved August 10th, 2021]

To solve this issue, each character has been manually replaced. Although it is not recommended to handle encoding problems in this manner, because errors might be missed or introduced, the small size of the review corpus allowed to solve the issue through this expedient with a satisfactory result.

Plain text versions (see Annex 2)

In the event that in the first series of experiments the selected taggers are not able to process XML corpora, an equivalent version in the plain text format have been created for Spoken, Literature, Review and Law.
The only difference between the XML and TXT versions is the replacement of all tags with placeholders, except for the sentence tags <s> and </s>. These placeholders consist of the symbol hash "#" immediately followed by the value of the attribute "name" of the elements <section> and <subsection> without any space in-between, as shown in the example below (Table 10):

XML version (tags)	TXT version (placeholders)
<section name="Title">	#Title
<section name="assemblee_sar_08">	#assemblee_sar_08
<subsection name="Article Text">	#ArticleText

Table 10. Conversion from XML files to TXT files

In addition to these tags, the Law corpus has also a <text> tag with a series of pairs attribute-value framing the contextual information of the *law* or *agreement*, which are the two types of legal texts present in the corpus. Moreover, <s> tags have an attribute *id* whose value defines the type of text (*law* or *agreement*) and its number, the language version, and the sentence's number. For convenience, the text tag has been replaced with the information extracted from the *id* attribute of the sentence tags, such as the type of text and its number. Table 11 provides an example:

XML version (tags)	TXT version (placeholders)
<text date_entry="1 janvier 1987" date_signature="14 juillet 1986" date_status="NA" decade_entry="1980" id="0.632.401.813" original_text="Y" topic_macro="0.6 Finances" topic_micro="0.63 Douanes" type="agreement" url="https://www.admin.ch/opc/fr/classified-compilation/19860201/index.html"> <section name="Title"> **<s id="agr023_fr_1">**	#Agreement023

<text date_entry="NA" date_signature="17 décembre 2004" date_status="NA" decade_entry="NA" id="NA" original_text="NA" topic_macro="0.6 Finances" topic_micro="NA" type="law" url="https://www.admin.ch/opc/fr/official-compilation/2005/2185.pdf"> <section name="Title"> **<s id="ord114_fr_1">**	#Law114

Table 11. Conversion specific to the Law corpus

As far as the sentence tags <s> and </s> are concerned, since all four corpora were organised according to the format "one sentence per line", it was sufficient to delete the sentence tags without compromising the structure of the files. Finally, a blank line has been added between each sentence to ease sentence recognition in the outputs of MElt and UDPipe whereas the blank line is substituted with a placeholder, such as "#s", for the processing with TreeTagger since this system automatically eliminates from the input file any blank line.

3.2.2 Evaluation Corpus (2nd phase)

The present section accounts for the design and construction of the Evaluation corpus (see Annex 2) that will serve in the second round of experiments to assess the performance of the taggers on the various text typologies.

The Evaluation corpus is composed of 50 sentences randomly extracted from each one of the four corpora (Spoken, Literature, Review, and Law) which were described in the previous section (3.2.1). Table 12 gives and account of the number of words and tokens of the four sub-corpora forming the Evaluation corpus, each one of them corresponding to a text typology: since the size of the corpus is moderate the tokens count has been carried out manually while the word counts has been carried out on SketchEngine.

SUB-CORPUS	WORDCOUNT	TOKENS
Spoken	~ 678	687
Written (Literature)	~ 627	756
Written (Review)	~ 601	670
Written (Law)	~ 1,309	1,563
Total	**~ 3,215**	**3,676**

Table 12. Evaluation corpus: words and tokens count.

With respect to the proportion of this corpus, Table 12 shows an alarming imbalance in the number of words and tokens of the Law sub-corpus if compared to the other three sub-corpora: this is due to the fact that the Law sub-corpus contains longer sentences.

This factor does not represent an issue for the present evaluation since the main goal is to determine the POS tagging performance of the selected system on each one of the four text typologies available. However, since an overall score of the taggers' performance will be provided at the end of this thesis, this imbalance will be taken into account and levelled.

Considering the findings gathered during the first round of experiments, it became apparent that

- not all taggers are able to correctly handle corpora in an XML format; and
- all systems have an internal module for the tokenization of texts, but they do not operate tokenization in the same manner.

With respect to the file format, the problem is not the format per se, but rather the presence in the input file of tags starting with the symbol "<" and ending with ">". As a matter of fact, when running the three taggers, one system prompted an error as soon as it encountered those symbols "<" and ">", while another one treated all XML tags as normal tokens: it separated symbols and punctuation marks from the sequences of letters and annotated all of them separately.
Hence, to accommodate all POS taggers and to avoid any discrepancy between the annotated corpora in terms of number of tokens, it was decided that the Evaluation corpus

- is stored in a plain text format with UTF-8 encoding,
- contains no tags but rather placeholders (consisting of two hash symbols "##" immediately followed by the name of the sub-corpus) to keep track of the text typologies, and
- is arranged in one-token-per-line with a blank line between each sentence to mark its boundaries. Sentence boundaries are marked by blank lines only for MElt and UDPipe, while for TreeTagger the blank lines are replaced by a placeholder such as "#s". This is done in order to distinguish one sentence from the other because TreeTagger automatically eliminates any blank line found in the input file.

Since the layout of the Evaluation corpus is arranged in one-token-per-line, few expedients have been adopted with regard to specific tokenization challenges (discussed

in section 1.2.1) in order to be able to evaluate the outputs of the three taggers on a common ground. Tokenization challenges are therefore addressed as follows:

1- multiword expression such as coordinating and subordinating conjunction, complex preposition, adverbial phrases, time expressions, etc., are not considered as single tokens since MElt is the only systems capable of correctly annotating all these entities,

2- named entities such as proper nouns and price values, are not considered as single tokens since the annotation of these elements is usually tackled with a tailor-made approach using specific tags,

3- cardinal numbers containing blank spaces and all items forming geographic coordinates have been kept together as a single token,

4- shortened words followed by a period, that is abbreviations, are merged as a single token,

5- numbered lists and the following punctuation mark are merged as a single token, such as, "II." "a)" and "(1)" (specificity of legal texts), and

6- clitics are split from the preceding word because in the training corpora used by MElt and TreeTagger clitics were segmented.

Table 13 summarize the expedients used to deal with specific tokenization challenges:

N°	TOKENIZATION CHALLENGE	ONE-TOKEN-PER-LINE ARRANGEMENT	
1	multiword expressions	à	Il
		cause	y
		de	a
2	cardinal numbers and geographic coordinates	47°24'	100 000
		de	hl
		latitude	
3	named entities such as proper nouns and price values	70	Royaume
		€	d'
			Espagne
4	abbreviations	Art.	par.
5	numbered lists and the following punctuation mark	III.	-1
6	clitics	concours	avait
		-là	-il

Table 13. Conventions on tokenization challenges

3.3 Gold-standard Corpus

The quantitative evaluation conducted in the second series of these experiments demands a reference corpus against which the Evaluation corpus can be compared, once it has been annotated by each one of the three taggers. This section accounts for the construction of the Gold-standard used for comparison.

The fact that the systems under consideration are trained on different corpora and lexical resources, means that they are likely to use different labels to annotate parts of speech. As we will see in section 3.4.2 which presents the systems' tagsets, this is the case. Therefore, a common tagset must be chosen to allow for the comparison of the taggers' outputs. The decision was made to adopt the Universal Dependency tagset (Nivre et al., 2016) which is used by UDPipe: this set contains 17 part-of-speech labels as shown in Table 14:

TAG	description	TAG	description
ADJ	adjective	PART	particle
ADP	adposition	PRON	pronoun
ADV	adverb	PROPN	proper noun
AUX	auxiliary	PUNCT	punctuation
CCONJ	coordinating conjunction	SCONJ	subordinating conjunction
DET	determiner	SYM	symbol
INTJ	interjection	VERB	verb
NOUN	noun	X	other
NUM	numeral		

Table 14. Gold-standard Corpus Tagset

The tagsets of MElt and TreeTagger are thus mapped to this tagset (see section 3.4.2).

Different strategies have been adopted to annotate the 200 sentences forming the Evaluation corpus and are presented hereafter:

- The 50 sentences of the review and literature corpora have been automatically annotated using UDPipe 2.0 (web application). Following this automatic annotation stage, the output has been manually checked and corrected.

- The 50 sentences of the Spoken corpus have been extracted from the TCOF-POS corpus which is part of PERCEO[12], a project aiming to design a morphosyntactic tagger for the annotation of spoken and written French corpora. In 2012, the TCOF-POS was the first freely available corpus of spontaneous spoken French with morphosyntactic annotation, that is part-of-speech and lemmas (Benzitoun at al., 2012). Once the 50 sentences have been extracted, the tags used in the TCOF-POS have been converted into the one's of the Universal Dependency tagset.

- The 50 sentences of the law corpus have been extracted from the morphosyntactic annotated version of the CHEU-lex's French sub-corpus. The tags used in the French version of CHEU-lex have been mapped into the Universal Dependency tagset afterwards.

The fact that there is just one human annotator participating in the consistency checking and manual correction of the POS annotation of the Gold-standard corpus, and in the comparison of the latter against the annotated outputs, is clearly a major drawback. To overcome this limit and reduce the subjectivity (bias) deriving from the lack of consultation with other annotators, different strategies have been adopted: already annotated corpora have been used when possible and the choice of a POS tag has been mostly determined according to the annotation guidelines created for the corpus PERCEO and provided with it. These have been elaborated from the annotation guidelines of the FTB (Abéillé & Clémenet, 2006).

3.4 Selected taggers

Three factors have influenced the selection process of the POS taggers to be evaluated: the choice was primarily oriented towards those systems that are freely accessible for non-commercial use, thus platforms like SketchEngine or Watson NLU[13] have been excluded a priori. Given a non-exhaustive list of potential POS taggers (see Annex 3), only the systems equipped with a French language model have been retained: this second criterion reduces inevitably the range of POS taggers available because it assumes that the systems have already been trained or at least conceived for the annotation of French

[12] https://www.ortolang.fr/market/corpora/perceo [Retrieved February 24th, 2021]
[13] https://www.ibm.com/cloud/watson-natural-language-understanding [Retrieved February 24th, 2021]

corpora. The third criterion is based on the ease of implementation: natural language toolkits and libraries have been crossed out because they are deemed to require a good command of a programming language to be utilised. Thus, the priority has been given to systems with a relatively simple implementation: POS taggers provided with a graphical user interface or with a web-based interface are considered more accessible and convenient for users who are not comfortable working by command-line.
The taggers shortlisted for the present evaluation are given hereafter:

1. MElt 2.0b12[14]
2. TreeTagger[15]
3. UDPipe 2.0 UD 2.6[16]

The following tables (Tables 15, 16 and 17) provide an account of some past research in computational linguistics that have evaluated the performance of the selected taggers on French corpora.

MElt 2.0b12 (Denis & Sagot, 2012)	
Training Corpus + Lexicon	POS accuracy
a variant of FTB + morphosyntactic information from Lefff	97.75%

Table 15. Accuracy score for MElt.

TreeTagger	
Training Corpus	POS accuracy
French MULTITAG (Allauzen & Bonneau-Maynard,	95.70%
a variant of FTB (Denis & Sagot, 2012)	96.12%

Table 16. Accuracy scores for TreeTagger.

[14] http://almanach.inria.fr/software_and_resources/custom/MElt-en.html [Retrieved March 23rd, 2021]
[15] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ [Retrieved January 15th, 2021]
[16] UDPipe (cuni.cz) [Retrieved March 23rd, 2021]

UDPipe 2.0 (Straka, 2018)		
Training Corpus	UPOS (= Universal part-of-speech) accuracy	Lemma accuracy
French-GSD	96.32%	96.75%
French-Sequoia	97.56%	97.36%
French-Spoken	95.47%	95.98%

Table 17. Accuracy score for UDPipe 2.0.

A brief account of the annotation strategy of the selected taggers is summarised in 3.4.1 whereas a glimpse on their internal tagsets is given in 3.4.2.

3.4.1 Tagging approaches

A brief description of the tagging approach used by MElt v2.0b12 (Denis & Sagot, 2012), TreeTagger (Schmid, 1994a) and UDPipe 2.0 (Straka, 2018) is provided hereafter. Going back to the binary classification of POS tagging approaches given in section 1.3, we find that all these systems fall into the category of the data-driven.

MElt

MElt is a Python implementation of a Maximum Entropy Markov Model (MEMM) also MaxEnt Markov Model. A MEMM combines features of a Hidden Markov model (HMMs described in section 1.3) and a maximum entropy (MaxEnt) model. As Denis & Sagot (2012, p. 3-4) claim, "an important appeal of MaxEnt models is that they allow for the combination of very diverse, potentially overlapping features without assuming independence between the predictors". As a matter of fact, MElt has the ability to use information which are extracted from both a training corpus and an external morphological lexicon. The training corpus for French is the French Treebank (Abeillé et al., 2003) while the lexicon used is *Lefff* (Sagot, 2010). Lefff is a large coverage lexicon that contains both morphological and syntactic information, even though only the morphosyntactic information are exploited in MElt.

In this type of model, a sequence of tags is assigned to a given sequence of words by means of a maximum likelihood estimation (MLE). Moreover, "the choice of the parameters is subject to constraints that force the model expectations of the features to be equal to their empirical expectations over the training data" (Denis & Sagot, 2012, p. 4).

An advantage of MaxEnt is the fact of being very fast to train.

TreeTagger

TreeTagger is a probabilistic part-of-speech tagger that uses Decision Trees (DTs). It is classified as a data-driven approach which also uses a series of rules to determine the correct tag. As a matter of fact, the goal of decision trees classifiers is to create a training model that can be used to predict the class or value of target variables by learning simple decision rules inferred from the training data (Jijo & Mohsin Abdulazeez, 2021), such as an already annotated corpus. The lexicon paired with the tagger contains the a priori tag probabilities for each word and is divided in three parts: a fullform lexicon, a suffix lexicon, and a default entry (Schmid, 1994a).
This system models the probability of a tagged sequence of words recursively, like a second order Markov model, but it differs from the latter because it uses a binary decision tree for the estimation of transition probabilities (Schmid, 1994a). In this model, the best tag sequence for a given sequence of words is determined with the Viterbi algorithm.

UDPipe 2.0

UDPipe 2.0 (Straka, 2018) is a Python prototype that utilizes an artificial neural network with a single joint model to perform POS tagging, lemmatisation, and dependency parsing. It is trained only on CoNLL-U data and on pretrained word embeddings. The 2.0 version reuses the modules for tokenization, sentence segmentation and multiword token splitting from UDPipe 1.2 (Straka & Straková, 2017). For POS tagging, Straka (2018, p.198) argues that a straightforward model has been applied "first representing each word with its embedding, contextualizing them with bidirectional RNNs, and finally using a softmax classifier to predict the tags".
Despite having demonstrated that deep neural networks achieved state-of-the-art results in many NLP areas like POS tagging (Straka, 2018, p.198) UDPipe 2.0 models require more computation power.

3.4.2 Tagsets

A tagset is a collection of labels which represent word classes (Horsmann et al., 2015) or parts of speech. As mentioned in 1.1, there is no universal agreement on the number and level of detail (granularity) of part-of-speech tags for a given language. Since the taggers

are used in both series of experiments without being trained on a common corpus, their tagsets are different. A glimpse of the internal tagset of each tagger is provided below.

MElt

The current tagset used by MElt (Table 18) contains 29 tags for parts of speech (Crabbé & Candito, 2008):

TAG	description	TAG	description
ADJ	adjective	P	preposition
ADJWH	interrogative adjective	P+D	preposition + determiner amalgam
ADV	adverb	P+PRO	preposition + pronoun amalgam
ADVWH	interrogative adverb	PONCT	punctuation mark
CC	coordination conjunction	PREF	prefix
CLO	object clitic pronoun	PRO	full pronoun
CLR	reflexive clitic pronoun	PROREL	relative pronoun
CLS	subject clitic pronoun	PROWH	interrogative pronoun
CS	subordination conjunction	V	indicative or conditional verb form
DET	determiner	VIMP	imperative verb form
DETWH	interrogative determiner	VINF	infinitive verb form
ET	foreign word	VPP	past participle
I	interjection	VPR	present participle
NC	common noun	VS	subjunctive verb form
NPP	proper noun		

Table 18. MElt tagset.

TreeTagger

The current tagset used by TreeTagger (Table 19) contains 33 tags for parts of speech (Achim Stein, 2003):

TAG	description	TAG	description
ABR	abbreviation	PRP:det	preposition + article
ADJ	adjective	PUN	punctuation
ADV	adverb	PUN:cit	punctuation citation
DET:ART	article	SENT	sentence tag
DET:POS	possessive pronoun	SYM	symbol
INT	interjection	VER:cond	verb conditional
KON	conjunction	VER:futu	verb futur
NAM	proper name	VER:impe	verb imperative
NOM	noun	VER:impf	verb imperfect
NUM	numeral	VER:infi	verb infinitive
PRO	pronoun	VER:pper	verb past participle
PRO:DEM	demonstrative pronoun	VER:ppre	verb present participle

PRO:IND	indefinite pronoun	VER:pres	verb present
PRO:PER	personal pronoun	VER:simp	verb simple past
PRO:POS	possessive pronoun	VER:subi	verb subjunctive imperfect
PRO:REL	relative pronoun	VER:subp	verb subjunctive present
PRP	preposition		

Table 19. TreeTagger tagset.

UDPipe 2.0 UD 2.6

The current tagset used by UDPipe 2.0 (Table 20) is the Universal Dependencies (UD) Tagset version 2.6 (Nivre et al., 2016) which contains 17 tags for parts of speech and 24 tags for morphological features[17]:

TAG	description
ADJ	adjective
ADP	apposition
ADV	adverb
AUX	auxiliary
CCONJ	coordinating conjunction
DET	determiner
INTJ	interjection
NOUN	noun
NUM	numeral
PART	particle
PRON	pronoun
PROPN	proper noun
PUNCT	punctuation
SCONJ	subordinating conjunction
SYM	symbol
VERB	verb
X	other

Lexical features	Inflectional features	
PronType	Gender	VerbForm
NumType	Animacy	Mood
Poss	NounClass	Tense
Reflex	Number	Aspect
Foreign	Case	Voice
Abbr	Definite	Evident
Typo	Degree	Polarity
		Person
		Polite
		Clusivity

Table 20. UDPipe 2.0 tagset: POS tags and morphological features.

The fact that the systems under consideration use different labels to annotate parts of speech means that a common tagset must be chosen to allow for the comparison of the taggers' outputs. Common practice suggests that it is better to map fine-grained tagsets on coarse grained tagsets, even though subtle distinctions are inevitably lost in the process (Horsmann et al., 2015). The tagset chosen is the Universal Dependency (Nivre et al., 2016), that is the tagset used by UDPipe, since it is the most coarse-grained out of the three as the morphological features are not taken into account except for the one

[17] For more information about the lexical features and their value, see https://universaldependencies.org/u/feat/index.html [Retrieved May 9th, 2021]

describing the "mood", "tense" and "form" of verbs. The mapping of the tagsets has been carried out by means of regular expressions on Notepad++,

Table 21 shows the correspondences between the different tagsets: the coloured cells highlight the lack of a straightforward match between tagsets for a specific label. However, MElt and TreeTagger do annotate some of these parts-of-speech even if it is by means of a different tag: in each cell, the overlapping tag is provided (except for "AUX" and "VERB" for which the correspondences are shown in a separate table, that is Table 22). As far as the evaluation is concerned, the tags in the coloured cells are marked as correct if they are assigned by MElt and TreeTagger according to the grammatical nature of the token and according to their position in the context.

UD Tagset		MElt Tagset	TreeTagger Tagset
ADJ	adjective	ADJ, ADJWH	ADJ
ADP	adposition	P, P+D, P+PRO	PRP, PRP:det
ADV	adverb	ADV, ADVWH	ADV
AUX	auxiliary	**(see Table 22)**	**(see Table 22)**
CCONJ	coordinating conjunction	CC	**KON**
SCONJ	subordinating conjunction	CS	
DET	determiner	DET, DETWH	DET:ART, DET:POS
INTJ	interjection	I	INT
NOUN	noun	NC	NOM
NUM	numeral	**DET, ADJ, NOUN, etc.**	NUM
PART	particle (-t)	-	-
PRON	pronoun	PRO, PROREL, PROWH, CLO, CLR, CLS	PRO, PRO:DEM, PRO:IND, PRO:PER, PRO:POS, PRO:REL
PROPN	proper noun	NPP	NAM
PUNCT	punctuation	PONCT	PUN, PUN:cit, SENT
SYM	symbol	-	SYM
VERB	verb	**(see Table 22)**	**(see Table 22)**
X	other	ET, PREF	ABR

Table 21. Correspondences between tagsets.

If we look at Table 21, it becomes clear that TreeTagger does not distinguish between subordinating (SCONJ) and coordinating conjunction (CCONJ). However, it was decided not to penalise the system for this lack of detail, which means that the "KON" tag is accepted as equivalent for both tags present in the UD tagset, namely "CCONJ" and "SCONJ". On the contrary, some other tags are totally absent from the tagsets used by TreeTagger and MElt. These are:

- “DET:dem” standing for demonstrative determiners in TreeTagger,
- “NUM”, numbers, and “SYM”, symbols, in MElt.

Again, it was decided not to penalise the systems and to consider valid the tags appended as long as they are correctly annotated with respect to the context in which the respective tokens are found. An example of the equivalences that have been accepted for the POS tag “NUM” (numbers) is provided below:

Goldstandard			MElt		TreeTagger		UDPipe	
TOKEN	TAG	LEMMA	TAG	0|1	TAG	0|1	TAG	0|1
une	PRON	un	DET	0	NUM	0	PRON	1
de	ADP	de	ADP	1	ADP	1	ADP	1
six	NUM	six	PRON	0	NUM	1	NUM	1
équipes	NOUN	équipe	NOUN	1	NOUN	1	NOUN	1
…			…		…			
le	DET	le	DET	1	DET	1	DET	1
8	NUM	8	DET	1	NUM	1	NUM	1
octobre	NOUN	octobre	NOUN	1	NOUN	1	NOUN	1
1986	NUM	1986	NOUN	1	NUM	1	NUM	1
…			…		…			
dans	ADP	dans	ADP	1	ADP	1	ADP	1
la	DET	le	DET	1	DET	1	DET	1
première	ADJ	premier	ADJ	1	NUM	0	ADJ	1
division	NOUN	division	NOUN	1	NOUN	1	NOUN	1
…			…		…			
le	DET	le	DET	1	DET	1	DET	1
1er	NUM	1er	ADJ	1	NUM	1	NUM	1
janvier	NOUN	janvier	NOUN	1	NOUN	1	NOUN	1
1987	NUM	1987	NOUN	1	NUM	1	NUM	1

Table 22 shows in more detail the correspondences for the tags auxiliary (AUX) and verb (VERB) along with the morphological information retained to ensure the highest degree of equivalence between tagsets.

UD			MElt	TreeTagger
TAG: *AUX*			TAG	TAG
Mood	*Tense*	*VerbForm*		
Ind	Pres	Fin	V	VER:pres
	Imp	"		VER:impf
	Futu	"		VER:futu
	Past	"		VER:simp
Sub	Pres	"	VS	VER:subp
	Imp	"		VERsubi
Cnd	Pres	"	V	VER:cond
/	Pres	Part	VPR	VER:ppre
Imp	Pres	Fin	VIMP	VER:impe
/	Pres	Inf	VINF	VER:infi
TAG: VERB			TAG	TAG
Mood	*Tense*	*VerbForm*		
/	Past	Inf	VPP	VER:pper

Table 22. Correspondences for the tag VERB and AUX.

4. Experiments

The present chapter accounts for the two series of experiments on three morphosyntactic taggers – MElt, TreeTagger, and UDPipe 2.0 – for French. The approach to POS tagging taken in these experiments is raw and unpretentious, but it is the one most likely adopted by non-experts.

Let us look again at the tasks which are going to be performed in each series:

First phase of experiments (4.1)

Taggers are tested on four corpora (Spoken, Literature, Review, and Law) given in an XML format. If the XML corpora cannot be processed, their equivalent plain text versions are used. The taggers operate the segmentation of sentences into tokens, annotate the resulting tokens with POS tags and provide a lemma for each token.

Second phase of experiments (4.2)

The taggers are run on the Evaluation corpus and set to carry out only the annotation in parts of speech. Their outputs are then compared against a reference corpus, namely the Gold-standard.

The first round of experiments (4.1) is useful to assess

- the presence or absence of the modules that are usually embedded in the architecture of POS tagging systems – tokenization and lemmatization – along with their strengths and weaknesses (characteristic of *Efficiency*, see Quality model 3.1)
- the degree of user-friendliness of the systems' interfaces (sub-characteristic of *Context completeness*, see Quality model 3.1). An account of the procedures needed for their implementation is also given.

The second round of experiments (4.2) allows to assess the performance of the system in the POS tagging task: the output of each tagger is compared against the Gold-standard and an accuracy score for each text typology is computed (characteristic of *Effectiveness*, see Quality model 3.1).

4.1 First phase

The focus of these first experiments is not on the performance of the tagger in relation to part-of-speech tagging but rather on:

- the user-friendliness of each tagger with respect to the setup phase and the deployment phase which is carried out through their respective users' interfaces (*Context completeness*)
- the availability of embedded modules which participate in the annotation, and the system ability to deal with different layouts and file formats (*Efficiency*).

To gather information pertaining to the evaluation of the quality in use sub-characteristic of *Context completeness* and characteristic of *Efficiency,* a black-box usage is simulated with each one of the taggers. This allows to understand which system has the most user-friendly interface, which modules are integrated, which layouts are handled, and which file formats can be used as input.

I start with a detailed description of the setup and deployment of each tagger (4.1.1), then I move onto a summarization of the modules available and of the layout and file formats handled (4.1.2).

4.1.1 Context completeness

In this section, I will attempt to provide an exhaustive account of the necessary steps for the installation (if required) and deployment of each one of the selected taggers. However, the only factor that is ultimately taken into account for the evaluation of the context completeness sub-characteristic is the presence or absence of a user-friendly interface by which it is possible to confidently interact with the system. With the term "interface" I refer to "a connection between two pieces of electronic equipment, or between a person and a computer"[18]. An interface can therefore be: (1) a graphical-user interface (GUI), (2) a web-based interface as for a web application, and (3) a command-line interface. The most user-friendly interfaces are estimated to be (1) and (2) as defined in section 3.1.

[18] https://dictionary.cambridge.org/dictionary/english/interface [Retrieved June 1st, 2021]

MElt 2.0b12

SETUP

MElt tagger runs on UNIX operating systems. One way to run MElt on a Windows OS, such in this case, is to install Cygwin and type any command through its console. Cygwin[19] is a programming and runtime environment, which allows source code designed for UNIX-like operating systems to be compiled with minimal modification and executed. Packages[20] must be installed from Cygwin to successfully run the tagger, among which python 2.7 (newer version such as the 3.8 presented problems with the syntax of some of the tagger's modules), pip, Perl, the libraries "NumPy" and libiconv, Perl-DBI (to use the lemmatizer), and the Cygwin command "make".

After downloading the tagger into the local disk directory C:\, I use the Cygwin terminal to configure it and install it, following the instruction of the user manual provided along with the tagger. It is worth mentioning at this point that to run the tagger I had to transfer the content of the folder \MEltTagger\pkgpythonlib into the folder \MEltTagger\bin. The reason seems to be a problem with the import of modules located in the "pkgpythonlib" folder that the python script of the tagger could not find when they are located outside the folder "bin".

DEPLOYMENT

A screenshot of Cygwin command-line interface is shown hereafter.

Figure 5. MElt on Cygwin command-line interface.

[19] https://cygwin.com/ [Retrieved May 10th, 2021]
[20] https://cygwin.com/packages/package_list.html [Retrieved May 10th, 2021]

Figure 5 shows an example of a command which asks MElt to normalise (-N) POS tag (-T) and lemmatise (-L) an input file called "spokenTest.txt" and to return an output file called "spokenMEltPOS.txt":

$ cat spokenTest.txt | /MEltTagger/bin/MElt -N -T -L > spokenMEltPOS.txt

Various options for tagging and lemmatising are available within MElt in addition to the ones used above, "-T" and "-L" as stated in the user manual, however I have found that these two are the most appropriate for the corpora in hand. One of the advantages of MElt is this normalisation option "-N" which enables the system to process noisy data, such as web texts or spoken transcripts. A second advantage is the ability of MElt of labelling unconventional data such as email addresses, URLs and emojis: the system automatically does so without users having to call a particular option for these instances.

TreeTagger

SETUP

TreeTagger successfully operates on Windows OS system. Installation packages are available for several other systems, such as PC-Linux, Mac-OS, and ARM. Parameters files for French are provided by Achim Stein[21], thus they must be downloaded and saved into TreeTagger\lib directory. Treetagger can be run by means of a graphic interface (GI), which is designed and maintained by Ciarán Ó Duibhín[22], or by command line. The graphic interface operates only on Windows systems and must be downloaded and saved into the TreeTagger\bin folder.

DEPLOYMENT

A screenshot of TreeTagger interface for Windows is shown hereafter (Figure 6):

[21] https://sites.google.com/site/achimstein [Retrieved January 15th, 2021]
[22] http://www.smo.uhi.ac.uk/~oduibhin/oideasra/interfaces/winttinterface.htm [Retrieved January 15th, 2021]

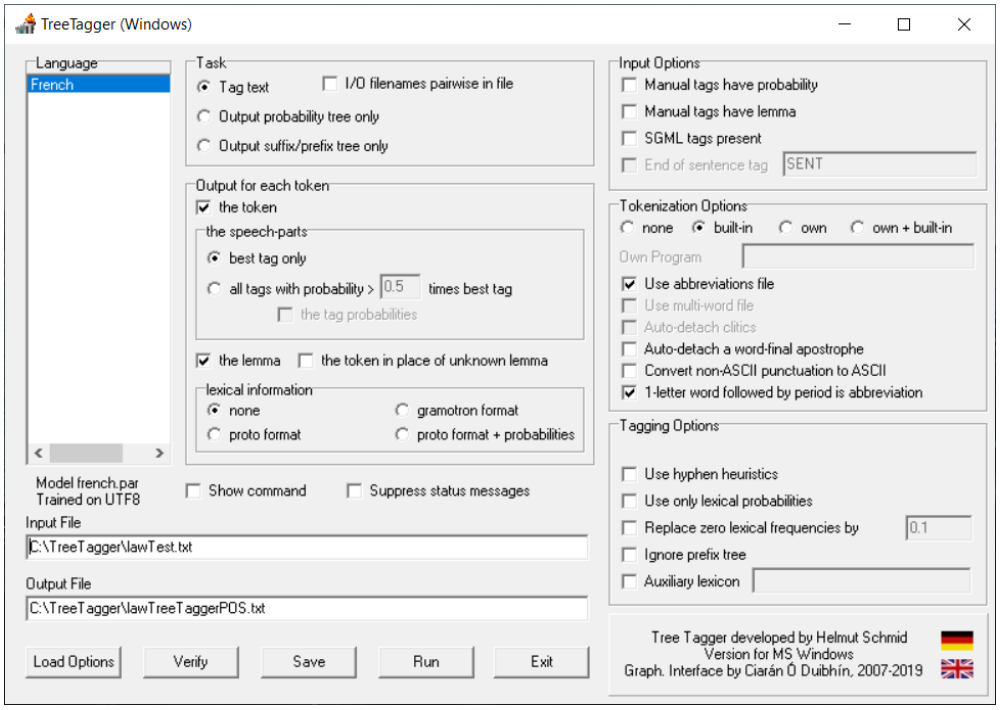


Figure 6. TreeTagger through graphic interface.

Figure 6 shows the options available on the TreeTagger Interface: I will not go through them since they are all described in detail in the README file provided with the system. The same options are also available by command line: in this case, the tagger is used through the Windows command prompt.

UDPipe 2.0 UD 2.6

SETUP

UDPipe 2.0 is available as a binary for Linux/Windows/OS X, as a library for C++, Python, Perl, Java, C#, and as a web service. During the first and second round of experiments, only the web application has been tested.

UDPipe 2.0 can be directly accessed through an internet browser at the following link: https://lindat.mff.cuni.cz/services/udpipe/. A wide range of Universal dependencies (UD) corpora in several different languages are made available for the training of the 2.0 model; they are also frequently updated. The training corpus used for the experiments with UDPipe 2.0 is the integrated French-GSD 2.6.

DEPLOYMENT

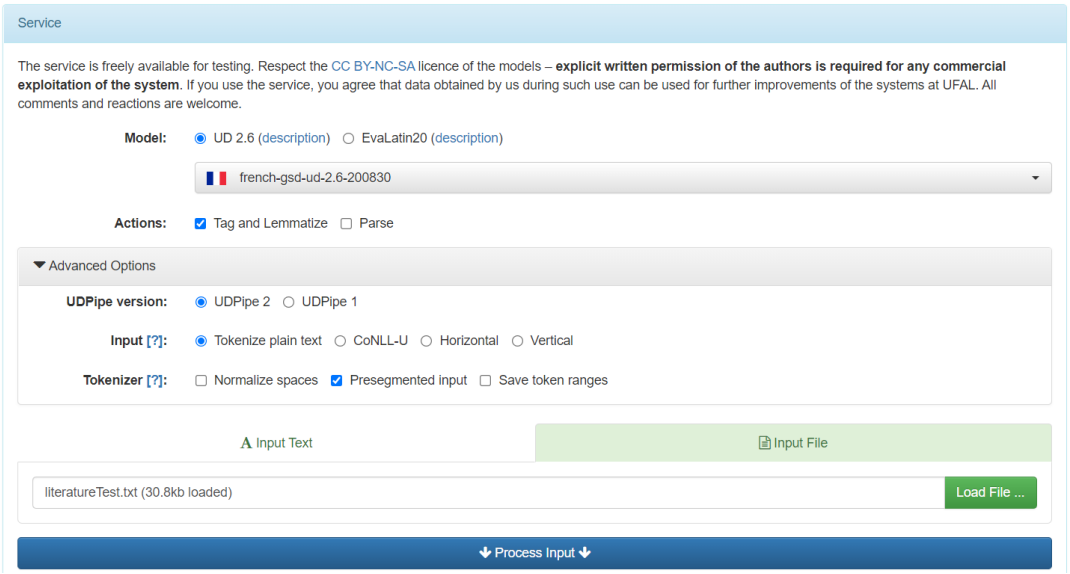


Figure 7. UDPipe 2.0 web application interface.

According to the corpus layout, UDPipe 2.0 has several options that can be checked as shown in Figure 7. If users hover the mouse over an option, information regarding that option are provided.

4.1.2 Efficiency

The *efficiency* of a system is given by the "resources expended in relation to the accuracy and completeness with which users achieve goals" (ISO/IEC, 2011, p.8). A part-of-speech tagger would be considered efficient in the current context if it includes all of the following modules: tokenization, POS tagging and lemmatization. With respect to tokenization, it is interesting to compare systems according to their ability to consider multiword expression (MWE) and to segment them appropriately (as illustrated in *Tokenization*, section 1.2.1). Regarding part-of-speech tagging, it is worth comparing systems according to the type of annotation provided, that is with POS tags or POS tags and morphological features. At last, given that electronic texts are nowadays stored in a variety of formats, the most used ones being XML and plain text, a system will be considered efficient if it can handle both these text formats.

MElt

MODULES

MElt embeds in its system all the required modules, namely tokenization, POS tagging and lemmatisation.

The tokenizer integrated in MElt correctly segments multiword expression (MWE) and annotates them with a specific part-of-speech. This is the only tagger out the three which is able to take into account MWE. However, if the clitics in the training corpus are separated from the preceding word, they must be segmented in the input corpus as well so that they can be correctly annotated with POS labels.

Concerning POS tagging, MElt has labels for parts-of-speech, but it does not have labels for morphological features.

Finally, with respect to lemmatisation, MElt provides a lemma for each token in the test corpora. The lemmas for words that are not present in the lexicon are preceded by a symbol "*".

LAYOUT

For an optimal annotation, the layout of the input file must be in the one sentence per line format. However, during the first testing phase MElt has been capable of segmenting also texts which are not arranged in this format.

FILE FORMATS

The user manual states that it is possible to annotate a corpus containing XML tags. This is true if users wish to perform only the part-of-speech tagging of the corpus: however, probably because the option -T has been used for tokenization, XML tags are assigned a random part-of-speech in the output file. On the contrary, Cygwin prompts an error (Figure 8) when users attempt to perform the annotation of a corpus with both parts of speech and lemmas:

```
Format error in lemmatizer input near to: /NC  at /usr/local/bin/MElt_lemmatiser.pl line 214, <> line 1.
Traceback (most recent call last):
  File "/MEltTagger/bin/MElt_tagger.py", line 1257, in <module>
    zh_mode=options.ZH)
  File "/MEltTagger/bin/MElt_tagger.py", line 433, in apply
    outfile.flush()
IOError: [Errno 32] Broken pipe
```

Figure 8. MElt lemmatizer error.

For this reason, during the rest of the experiments with MElt, I used the plain text versions of the four corpora discussed in section 3.2.1.

Finally, the output file produced has the same layout of the input file, that is one sentence per line, and is presented in the format "token/POS tag/lemma" as shown below:

> *Ça/PRO/cela ne/ADV/ne cole/V/*coler pas/ADV/pas à/P/à l'/DET/le évier/NC/évier .../PONCT/...*

du_coup/ADV/du_coup ça/PRO/cela ne/ADV/ne sert/V/servir à/P/à rien/PRO/rien

(See Annex 4)

TreeTagger

MODULES

Treetagger embeds in its system all the required modules, namely tokenization, POS tagging and lemmatisation. However, as explained in the section below (LAYOUT), the integrated tokenizer is not able to segment clitics and contractions containing apostrophe therefore users must foresee a pre-processing stage to manually split these elements so that they can be correctly annotated with POS labels. Concerning POS tagging, TreeTagger has labels for parts-of-speech, but it does not have labels for morphological features.

Finally, with regards to lemmatisation, if TreeTagger is unable to recognise the lemma from the training corpus or the integrated lexicon, it can either replace the lemma with the token or with an "<unknown>" element. This workaround is very useful since it provides a mean for users to easily detect this kind of problems and make appropriate corrections. When using the graphical interface, users are left to choose between token and "<unknown>" element. However, when using TreeTagger by command line, if the system is unable to find the lemma, the latter is replaced with the token even if the option to do so is not called as argument. The following example (Figure 9) taken from the review corpus shows how this "<unknown>" element works:

Token	POS	Lemma
Acheté	*VER:pper*	*acheter*
pour	*PRP*	*pour*
aller	*VER:infi*	*aller*
sur	*PRP*	*sur*
un	*DET:ART*	*un*
siege	*NOM*	*<unknown>*
Recaro	*NAM*	*<unknown>*

Figure 9. TreeTagger Output example

LAYOUT

TreeTagger has no restriction for what concerns text layouts: it accepts files with a conventional layout, for example divided in paragraphs, as well as files in one sentence per line. However, it is recommended to input files that are arranged in one token per line.

Since the tagger was trained on the French FTB as stated in the README file, it requires text-specific expressions to be separated from the preceding word by a blank space to provide a correct annotation: this is the case for clitics and contractions containing an apostrophe, as for example the determiner "*le*" that becomes "*l'*" when found in front of a word starting with a vowel, like "*abeille*". At last, every apostrophe in the input file must be converted into a straight single quote for the tagger to be able to recognise it. Failing to do so will result in the following output:

l'Assemblée NOM <unknown>

FILE FORMATS

TreeTagger can process both plain text and XML files containing metadata and structural tags. To avoid the annotation of the XML tags the corresponding option "SGML tags present" must be ticked in the GI or called as argument "-sgml" when running the tagger by command line.

Finally, as shown in Figure 9, the output of TreeTagger is arranged with one token per each line followed by a POS label and a lemma which are separated by a tab.

(See Annex 5)

UDPipe 2.0

MODULES

UDPipe 2.0 embeds in its system all the required modules, namely tokenization, POS tagging and lemmatisation.

The tokenizer integrated in UDPipe does not segment multiword expressions (MWEs) automatically.

Concerning POS tagging, UDPipe has labels for both parts-of-speech and lexical/ morphological features.

Finally, with respect to lemmatisation, UDPipe provides a lemma for each token in the test corpora.

LAYOUT

UDPipe 2.0 accepts different types of layouts as *Input*: one sentence per line, where each token must be separated from the preceding and following token by means of a white space (option *Horizontal*); one token per line (option *Vertical*); CoNLL-U layout (this

option serves when users aim to carry out parsing); and normal layout which is tokenized by the internal module of UDPipe (option *Tokenize plain text*).
With respect to the Horizontal Input option, it is worth mentioning that not only words but rather every token (punctuation included) must be separated with a white space. Failing to do so will result in punctuation marks attached to the preceding or following token as in the input text, because the tokenizer is not used.

FILE FORMATS
UDPipe 2.0 do not process XML files. Users might try to do so by means of the Horizontal Input option: if tags consisting of a unique word with no attributes are present and are separated from words with a blank space, XML tags are annotated with a POS tag. For more complex tags which contain several pairs attributes-values separated by a blank space, the annotation result is catastrophic. For this reason, during the rest of the experiments with UDPipe, I used the plain text versions of the four corpora discussed in section 3.2.1.

The uniqueness of UDPipe with respect to the other two systems is that it produces not only an annotation in parts of speech and lemmatization of the tokens, but also an annotation of the grammatical properties of each token as show in the example below (Figure 10):

# text = Les bandes ne sont pas réfléchissantes mais seulement brillantes					
1	Les	le	DET	_	Definite=Def|Gender=Fem|Number=Plur|PronType=Art
2	bandes	bande	NOUN	_	Gender=Fem|Number=Plur
3	ne	ne	ADV	_	Polarity=Neg
4	sont	être	AUX	_	Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin
5	pas	pas	ADV	_	Polarity=Neg
6	réfléchissantes	réfléchissant	ADJ	_	Gender=Fem|Number=Plur
7	mais	mais	CCONJ	_	_
8	seulement	seulement	ADV	_	_
9	brillantes	brillant	ADJ	_	Gender=Fem|Number=Plur

Figure 10. UDPipe Output example

(See Annex 6)

Considering the findings on the *Efficiency* of the taggers, it is possible to fill in the following tables: Table 23 shows the efficiency in terms of file formats, Table 24 shows the efficiency in terms of file layouts (which is not taken into account in the evaluation

but gives us an idea of the requirements with respect to this feature) and Table 25 shows the efficiency in terms of modules embedded in the systems.

EFFICIENCY (File formats)	MElt	TreeTagger	UDPipe 2.0 WEB VERSION
XML tags	No*	Yes	No
Plain text	Yes	Yes	Yes

Table 23. Efficiency: file formats

** Only possible if users wish to use only the POS tagging module.*

EFFICIENCY (File layouts)	MElt	TreeTagger	UDPipe 2.0 WEB VERSION
Characters adaptation (e.g., apostrophes)	No	Yes	No
Contractions and clitics pre-tokenization	Yes*	Yes	No
One sentence per line compulsory	No**	No	No

Table 24. Efficiency: file layouts

** Only clitics.*

*** Unable to recognise MWE if input file is arranged in one-sentence-per-line. However, it almost always attributes the appropriate tags to each word forming the MWE token*

EFFICIENCY (modules)		MElt	TreeTagger	UDPipe 2.0 WEB VERSION
Tokenization	Single token	Yes	Yes	Yes
	MWE	Yes	No*	No
Tagging	POS	Yes	Yes	Yes
	Morphological features	No	No	Yes
Lemmatization		Yes	Yes	Yes

Table 25. Efficiency: modules

** Only possible if a list of multiword expressions is provided to the tagger.*

4.2 Second phase

In this second round of experiments, the three taggers are run on the Evaluation corpus described in 3.2.2 which is arranged in one-token-per-line and stored in a plain text format. Taggers are instructed to annotate the Evaluation corpus only with POS tags. Their outputs are then compared against a reference corpus, namely the Gold-standard corpus presented in 3.3. An accuracy score is computed for each text typology and for each tagger to assess the characteristic of the Quality Model (section 3.1) of *Effectiveness.*

4.2.1 Effectiveness

The effectiveness of a system is defined as the "accuracy and completeness with which users achieve specified goals" (ISO/IEC, 2011, p.8). To assess the effectiveness of the taggers considered (MElt, TreeTagger, UDPipe 2.0), the Evaluation corpus is provided as input to the systems. When running the taggers, the tokenization option is not selected since the Evaluation corpus is already tokenized. The corpora annotated with part-of-speech tags by MElt and TreeTagger are then mapped onto the Universal Dependencies tagsets by means of regular expression on Notepad++. Finally, all outputs are compared against the Gold-standard.

Table 26 shows the accuracy score computed for each tagger and for each text typology:

Accuracy on Text Typology			
Text types	MElt	TreeTagger	UDPipe
Spoken	84.57%	89.67%	92.29%
Literature	91.67%	93.52%	98.02%
Review	86.12%	90.60%	95.67%
Law	89.57%	89.32%	89.83%

Table 26. Taggers' accuracy score on four different text typologies.

The best performing system across all text typologies is UDPipe 2.0. While looking at the overall picture, we realise that all systems had their best performance on the sentences of the Literary genre, probably because it is the most conventional text typology where syntax is neater and does not contain typos, and text-specific entities, such as

- abbreviations (art., al., etc.), alphabetized and numerical lists for the legal genre

- interjections and broken words such as “conc-“ (= concernant) for speech transcripts.

A curious fact about the performance of UDPipe with respect to this genre is that, despite its excellent ability to attribute the correct tags to parts of speech, it had no little difficulty in appropriately lemmatising archaic verb forms that were present in the sentences of the literary sub-corpus. Its most striking weakness is instead the annotation of auxiliary verbs in the Spoken and Review sub-corpora even if its performance had not been affected as much: on several occasions, the system has annotated the French verb “être” (=to be) as auxiliary even if that was not the case in the context in which the verb was found.

The second-best performance with respect to the text typologies is achieved on

- the Review sub-corpus for UDPipe and TreeTagger
- while for MElt is achieved on the Law sub-corpus, the one in which the two counterparts have had their worst.

Most surprising is the eight-percentage point fall of UDPipe on the sentences of the legal genre: the main challenges for UDPipe here have been the annotation of abbreviations, prepositions, and some determiners, such as “tout”. Other difficulties in the Law sub-corpus that have weighed on the performance of all the systems in general are the handling of nominal sentences where the first common noun is either tagged as a verb or as a proper noun, the annotation of proper nouns at the exception of country names, and the recognition of units of measurements (TreeTagger has not struggled too much in the annotation of the latter).

In general, the main weaknesses of TreeTagger are the erroneous annotation of determiners and adjectives as pronouns and the annotation of common nouns at the beginning of a nominal sentence as verbs. Moreover, the system is not consistent in the annotation of abbreviations but performs very well on numbers.

It is worth mentioning at this point that MElt's performance was penalized in general by the lack of tags for the annotation of symbols, abbreviations, and numbers which are found in high number in the sentences of both the Spoken and the Law corpus. However, regarding numbers, it was decided to consider valid the tags appended as long as they

are correctly annotated with respect to the context in which the respective tokens are found (see section 3.4.2).

A difficulty in handling interjections, which abound in the sentences extracted from the speech transcripts, is the reason why MElt performed so poorly on this text typology, despite having a POS label for their annotation. In a corpus of such a small size, the inability of MElt to append the tag "INTJ" to interjections have weighed heavily on its overall performance. Table 27 shows a sentence extracted from the spoken sub-corpus: the correct tag is provided along with the POS label attributed by the three systems. The cells containing the wrong POS label with respect to the one found in the Gold-standard are highlighted in pink:

Goldstandard			MElt	TreeTagger	UDPipe	
TOKEN	TAG	LEMMA	TAG	TAG	TAG	Lexical Feature
euh	INTJ	euh	PRON	INTJ	INTJ	
sinon	ADV	sinon	CCONJ	KON	ADV	
par	ADP	par	ADP	ADP	ADP	
rapport	NOUN	rapport	NOUN	NOUN	NOUN	
aux	ADP	au	ADP	ADP	DET	
festivités	NOUN	festivité	NOUN	NOUN	NOUN	
ben	INTJ	ben	PRON	ADV	ADV	
tout	PRON	tout	PRON	ADV	PRON	
s'	PRON	se	PRON	PRON	PRON	
est	AUX	être	V	VER:pres	AUX	Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin
très	ADV	très	ADV	ADV	ADV	
bien	ADV	bien	ADV	ADV	ADV	
passé	VERB	passer	VPP	VER:pper	VERB	Gender=Masc|Number=Sing|Tense=Past|VerbForm=Part

Table 27. POS tagging errors in the Spoken sub-corpus.

With respect to the Review sub-corpus which contains user generated content, there are a few interesting challenges that are worth discussing along with their outcomes: these are (1) sentences written in all capital letters, (2) emojis and (3) typos. For the first challenge I have calculate the error rate of each tagger in the annotation of two sentences written in capital letters containing respectively 22 and 23 tokens. The average error rates for the systems are 16% for UDPipe, 13% for TreeTagger and 40% for MElt. The nature of the errors for TreeTagger and UDPipe was similar to the ones observed in the

other sub-corpora, but for MElt it appears that the system "goes haywire" when dealing with sentences in all capital letters. (2) Emojis have been annotated correctly 50% of the times by UDPipe and 0% of the times by TreeTagger and MElt. However, the Evaluation corpus contains only 4 emojis and MElt does not have a POS tag for symbols or other non-parts-of-speech entities. Finally, it is interesting the way in which the systems deal with typos (3) like the three ones shown in Table 28 (the cells containing the wrong POS label with respect to the one found in the Gold-standard are again highlighted in pink):

Goldstandard			MElt	TreeTagger	UDPipe	
TOKEN	TAG	LEMMA	TAG	TAG	TAG	Lexical Feature
j	PRON	j	NOUN	NOUN	PRON	
ai	AUX	avoir	V	VER:pres	AUX	Mood=Ind|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin
cramé	VERB	cramer	VPP	VER:pper	VERB	Gender=Masc|Number=Sing|Tense=Past|VerbForm=Part
le	DET	le	DET	DET	DET	
telephone	NOUN	telephone	NOUN	NOUN	NOUN	
à	ADP|(AUX)	à	ADP	ADP	AUX	Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin
duré	VERB	durér	NOUN	NOUN	VERB	Gender=Masc|Number=Sing|Tense=Past|VerbForm=Part
8	NUM	8	DET	NUM	NUM	
moi	NOUN	mois	PRON	NOUN	PRON	

Table 28. Example of the annotation of typos in the Review corpus

The first case is the first-person singular pronoun "j" (=I, "*je*" in French) that should be followed by an apostrophe: UDPipe is the only tagger able to annotate this case correctly. The second one is the preposition "*à*" (=to) in place of the auxiliary verb "a" (=to have, which is the third-person singular of the simple present of the verb "*avoir*") and the pronoun "*moi*" (=me) in place of the common noun "*mois*" (=month). It is curious how UDPipe is capable of annotating correctly these two typos, probably because it made better use of the context surrounding these tokens.

It would seem, therefore, that there are some systems that perform better than others in certain text types, but since the size of the corpus for evaluation is rather small it is difficult to establish clear error patterns beyond those already listed.

We conclude this excursus on the results obtained in the second series of experiments for the evaluation of the characteristic of *Effectiveness* with the overall performance of each tagger on the entire Evaluation corpus (Table 29):

Overall system's accuracy	
UDPipe	93.90%
TreeTagger	90.79%
MElt	88.13%

Table 29. Systems' overall performance

5. Results and Discussion

In this section, the results obtained in the two sets of experiments are regrouped. They are organized according to the properties (characteristics and sub-characteristic) of the Quality in use model of the ISO/IEC 25010:2011 that constitutes the framework of the present evaluation and was described in detail in section 3.1.

Effectiveness

Table 30 reports the accuracy scores obtained by the taggers on the four sub-corpora forming the Evaluation corpus. Each sub-corpus corresponds to a specific text typology with its unique annotation challenges which were discussed in section 3.2.1.

EFFECTIVENESS			
Text type / Sub-corpus	MElt	TreeTagger	UDPipe
Spoken	84.57%	89.67%	92.29%
Literature	91.67%	93.52%	98.02%
Review	86.12%	90.60%	95.67%
Law	89.57%	89.32%	89.83%

Table 30. Effectiveness - POS tagging accuracy

UDPipe is undoubtedly the system that has achieved the best performance in all text types, especially in the Literature corpus where the performance has reached a 98.02% of accuracy. Compared to this system, TreeTagger's performance was 4.5 percent lower on this sub-corpus while, MElt's performance was lower by 6.35 percent.

If we look at the performance obtained in the other text types, it is clear that the taggers' effectiveness was not the same: the second-best performance for MElt is achieved on the Law sub-corpus while for TreeTagger is achieved on the Review sub-corpus. If we rank the performances of the taggers, we obtain Table 31:

PERFORMANCE	MElt	TreeTagger	UDPipe
1st best	Literature	Literature	Literature
2nd best	Law	Review	Review
3rd best	Review	Spoken	Spoken
4th best	Spoken	Law	Law

Table 31. Ranking of the taggers' performance on text typology

These results suggest that there are some systems which perform better than others for specific text types, although this is not extremely evident.

In any case, it must be remembered that the potential of the systems was not fully exploited. It might therefore be interesting to repeat these experiments and

- train the taggers on corpora which share the same characteristic of each sub-corpora
- use larger corpora

to see if their performance changes and how.

Efficiency

As a result of the experiments described in 4.1.2, the following table (Table 32) has been devised to give account of the modules embedded in each system and of the systems' ability to handle different file formats. Different systems provide users with different possibilities. The findings gathered in this table therefore could be used to choose a system that is better suited for different orientations.

PROPERTIES			Melt	TreeTagger	UDPipe
FILE FORMAT	XML		No[1]	Yes	No
	Plain text		Yes	Yes	Yes
MODULES	Tokenization	Single token	Yes	Yes	Yes
		MWE	Yes	No[2]	Yes[3]
	Tagging	POS	Yes	Yes	Yes
		Morphological features	No	No	Yes
	Lemmatization		Yes	Yes	Yes

Table 32. Efficiency of the taggers

[1] Only possible if users wish to use only the POS tagging module.

[2] Only possible if a list of multiword expressions is provided to the tagger.

[3] Does not tokenise MWEs properly but assigns the correct tag to them

Context coverage

The experiments described in 4.1.1 Context completeness have enabled us to understand the procedures necessary for the setup and deployment of the systems and to acknowledge the type of interfaces available for users. Using that description, the following table (Table 33) can be filled in:

CONTEXT COMPLETENESS				
Taggers	Graphical INTERFACE for Windows OS	Web INTERFACE	Command-line INTERFACE	working only on UNIX systems
Melt	No	No	Yes	Yes
TreeTagger	Yes	No	Yes*	No
UDPipe	No	Yes	Yes*	No

Table 33. Context completeness of the taggers

** Not used.*

As mentioned in section 4.1.1, it is only the type of user interface that determines, in this kind of evaluation, the user-friendliness of a system. However, since MElt requires a UNIX[23] system to run, it is necessary to take this aspect into account and include it in Table 33, which summarises the context completeness sub-characteristic.

Limitations

As far as the three properties of the quality in use model are concerned, there are a few limitations to the present evaluation which are worth discussing.

To begin with, the assessment of the "*context completeness*" sub-characteristic, that is meant here as the degree of user friendliness of the taggers' interface, has been assessed by just one participant. However, different users have different competencies and preferences. Therefore, as much as this assessment aims to be objective, it cannot be assumed that the results obtained from the evaluation of this property are representative of an entire category of users with moderate skills in computer science. Users are

[23] https://en.wikipedia.org/wiki/Unix [Retrieved July 20th, 2021]

therefore left with an overview of the interfaces available for the three taggers. These overviews help users in the process of deliberation before choosing their selected interface, which will enable them to choose the system that best suits their skills.

Secondly, the size of the corpora used to assess the "*effectiveness*" of the systems is quite modest. Larger annotated corpora might be better suited for the evaluation of the performance of the POS taggers. Moreover, as mentioned in section 2.3, the accuracy of different taggers is usually assessed by means of a comparative evaluation in computational linguistic research. To do so, the systems must be trained on the same corpus and lexical resources: part of the corpus is usually excluded from the training data to provide an unseen test set. This is not the case of the present evaluation because the taggers have been trained on different corpora.

As far as the annotation stage is concerned, there are two potential limits: firstly, there is the fact that the consistency checking, and correction of the Gold-standard corpus has been carried out by one annotator. Hence, it has not been possible to discuss the ambiguities observed in the texts and to choose with confidence one part of speech over another for ambiguous tokens. To overcome this problem the guidelines of the PERCEO corpus have been followed (see section 3.3) to verify the correctness and consistency of the Gold-standard corpus annotation. Nonetheless, this course of action might not have been sufficient. As a matter of fact, when more than one annotator is involved in an evaluation campaign, ambiguities are overcome by means of other, more objective, methods, like the inter-annotator agreement score (discussed in section 2.1). Secondly, there is the problem related to the choice of a coarse-grained tagset for comparison: when dealing with tagging systems that use different POS labels, a tagset must be chosen as a reference and all the others must be mapped onto it. During this process, it is not unusual to be confronted with a lack of correspondence between the labels of the different tagsets in exam. In the present study, the tagset of UDPipe which has the lowest number of POS tags (thus the more coarse-grained tagset) has been chosen as reference. However, it could have been more appropriate to create two versions of the Gold-Standard corpus, one of which would have been annotated with POS tags from the tagset with the highest level of granularity: in this way, it would have been possible to compare the accuracy score of the systems with a more fine-grained tagset and to verify if the choice of a

different tagset granularity may have any particular effect on the evaluation of the system's performance.

With respect to tagging performance, it is common knowledge that POS taggers can improve their accuracy if they are trained on a corpus similar to the one to be annotated or customised with external resources (Coden et al., 2005; Savary et al., 2019). Although this line of action was not followed in this evaluation because it was out of its scope, it should be admitted that the performance results obtained here have been undoubtedly influenced by this choice.

Finally, a great deal of morphosyntactic taggers is available today, thus I cannot completely rule out the possibility that despite not being considered in this thesis, others might have a relatively simple implementation. It might be interesting then, to conduct further research to assess the user-friendliness of other systems.

6. Conclusion

In this master thesis, I have presented an evaluation of three part-of-speech taggers for French, namely MElt, TreeTagger and UDPipe 2.0 web application, through the framework of the Quality in use model (ISO/IEC 25010:2011 Standard). For each system, three properties have been evaluated, namely "efficiency", "effectiveness" and "context completeness". The aim of this evaluation was

- to identify which part-of-speech tagger for French produces the best annotation for a specific text typology among samples of speech transcripts, literary texts, product reviews and legal texts (*effectiveness*),
- to establish which system has the most user-friendly interface considering that this evaluation addresses translation students and other researchers with moderate knowledge in computer science (*context completeness*), and
- to determine how comprehensive each system is with respect to the type of files that can process (XML and plain text files) and the modules (tokenization and lemmatization) that are embedded within it (*efficiency*).

Based on the results obtained, it appears that UDPipe is the system achieving the best performance on all four text typologies with the highest accuracy score, that is 98.02%, on a sample of literary texts and the lowest score, 89.93% of accuracy, on a sample of legal texts. Each tagger performed their best in the literary text typology. The most problematic text types are instead legal texts and speech transcripts for TreeTagger and UDPipe, whilst for MElt the problematic areas include speech transcripts and product reviews.

As far as the effectiveness characteristic of the Quality model is concerned, it is not possible to clearly establish the correlation between the performance of a part-of-speech tagger and a specific text typology. This could be due to the fact that a raw approach to POS tagging was adopted in this evaluation: a black-box usage has been simulated but the taggers' potential for customisation has not been exploited since the systems were not retrained for a specific text typology and parameters have not been tuned. As a matter of fact, the internal language models of each tagger were used.

Two taggers satisfy the context completeness sub-property, namely UDPipe 2.0 which allows users to interact with the POS tagging system through a web-based interface and TreeTagger for which a graphical user interface is made available.

Finally, as far as the efficiency characteristic of the Quality model is concerned, TreeTagger is the only system that can handle both XML and plain text files correctly. Since TreeTagger does not compromise the structure of the XML tags present in the texts and is able to annotate them appropriately, it is considered the most efficient tagger in terms of file format handling. Even though all systems embed both the tokenization and lemmatization modules, MElt is considered more efficient because it is the only POS tagger capable of tokenizing French multiword expression (MWE) satisfactorily and of annotating them with the correct POS label.

A summary of the results obtained on this evaluation of three part-of-speech taggers for French is provided in Table 34:

EVALUATION RESULTS				
PROPERTIES		MElt	TreeTagger	UDPipe 2.0 WEB APPLICATION
EFFICIENCY	**File Format (XML and plain text)**		x	
	Modules (tokenization of MWE and lemmatisation)	x		
EFFECTIVENESS				x
CONTEXT COMPLETENESS			x	x

Table 34. French POS tagging evaluation Results

Going forward it might be interesting to deepen this work both by testing further systems and by training those already used with external lexical resources or with annotated corpora of the same domain as the one of the test corpora.

7. References

Abeillé, A., Clément, L., & Toussenel, F. (2003). Building a treebank for French. In A. Abeillé (Ed.), *Treebanks*. Kluwer, Dordrecht.

Abeillé, A., & Clément, L. (2006). Annotation morpho-syntaxique. Version du 10 novembre 2006.

Adda G., Lecomte J., Mariani J., Paroubek P., & Rajman M. (1998). The GRACE French Part-of Speech Tagging Evaluation Task. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)* (Vol. 1, pp. 433-441). Granada, Spain.

Adda G., Lecomte J., Mariani J., Paroubek P., & Rajman M. (2000). Les procédures de mesure automatique de l'action GRACE pour l'évaluation des assignateurs de Parties du Discours pour le Français. In K. Chibout (Ed.), *Ressources et évaluation en ingénierie des langues*. (pp. 645-664). Paris: De Boeck.

Allauzen, A., & Bonneau-Maynard, H. (2008). Training and evaluation of POS taggers on the French MULTITAG corpus. LIMSI/CNRS. Univ Paris-Sud, Orsay.

Analyse et traitement informatique de la langue française - UMR 7118 (ATILF), Institut de l'information scientifique et technique - CNRS UPS76 (INIST), Laboratoire d'informatique de Paris Nord - UMR 7030 (LIPN) (2012). PERCEO : un Projet d'Etiqueteur Robuste pour l'Ecrit et pour l'Oral [Corpus]. ORTOLANG (Open Resources and TOols for LANGuage) - www.ortolang.fr, v1, https://hdl.handle.net/11403/perceo/v1.

Analyse et traitement informatique de la langue française - UMR 7118 (ATILF) (2020). TCOF : Traitement de Corpus Oraux en Français [Corpus]. ORTOLANG (Open Resources and TOols for LANGuage) - www.ortolang.fr, v2.1, https://hdl.handle.net/11403/tcof/v2.1

Assal, H., Seng, J., Kurfess, F., Schwarz, E., & Pohl, K. (2011). Semantically-enhanced information extraction. In *IEEE Aerospace Conference Proceedings.* (pp. 1-14).

Benzitoun, C., Karën, F., & Benoît, S. (2012). TCOF-POS : un corpus libre de français parlé annoté en morphosyntaxe. In *JEP-TALN 2012 - Journées d'Études sur la Parole et Conférence annuelle du Traitement Automatique des Langues Naturelles*. Grenoble, France.

Brants, T. (2000). TnT: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing (ANLC '00)* (pp. 224-231) Association for Computational Linguistics. USA. https://doi.org/10.3115/974147.974178

Brill, E. (1992) A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing* (p. 152-155). Association for Computational Linguistics. Trento, Italy. https://doi.org/10.3115/974499.974526

Brill, E. (1995) Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. In *Computational Linguistics* (Vol. 21(4), pp. 543-565).

Candito, M., & Constant, M. (2014). Strategies for contiguous multiword expression analysis and dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Long Papers, Vol. 1, pp. 743-753). Association for Computational Linguistics. Baltimore, Maryland.

Coden, A. R., Pakhomov, S. V., Ando, R. K., Duffy, P. H., & Chute, C. G. (2005). Domain-specific language models and lexicons for tagging. In *Journal of Biomedical Informatics* (Vol. 38(6), pp. 422-430). https://doi.org/10.1016/j.jbi.2005.02.009.

Constant, M. (2012). Mettre les expressions multi-mots au cœur de l'analyse automatique de textes : sur l'exploitation de ressources symboliques externes. (Traitement du texte et du document). Université Paris-Est.

Constant, M., Eryiğit, G., Monti, J., van der Plas, L., Ramisch, C., Rosner, M., & Todirascu, A. (2017). Multiword Expression Processing: A Survey. In *Computational Linguistics* (Vol. 43(4), pp. 837-892). Association for Computational Linguistics.

Crabbé, B., & Candito, M. (2008). Expériences d'analyse syntaxique statistique du français. In *Actes de la 15ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN'08).* (pp. 45-54). Avignon, France.

Denis, P., & Sagot, B. (2012). Coupling an annotated corpus and a lexicon for state-of-the-art POS tagging. In *Language Resources and Evaluation* (Vol. 46(4), pp. 721-736). Springer-Verlag

EAGLES. (1999). The EAGLES 7-step recipe. EAGLES Evaluation Working Group. [Retrieved July 2021 from https://www.issco.unige.ch/en/research/projects/eagles/ewg99/7steps.html]

Fitschen, A., & Gupta, P. (2008). Lemmatising and morphological tagging. In A. Lüdeling & M. Kytö (Eds.), *Corpus Linguistics. An International Handbook* (Vol.1, pp. 552-564). Berlin: Walter de Gruyter.

Giménez, J., & Márquez, L. (2004). SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal.

Green, S., Marneffe, M. C., Bauer, J., & Manning, C. (2011). Multiword expression identification with tree substitution grammars: A parsing TOUR DE FORCE with French. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP 2011) (pp. 725-735)

Gui, T., Zhang, Q., Huang, H., Peng, M. & Huang, X. (2017). Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 2401-2410).

Hirschman, L., & Mani, I. (2004). Evaluation. In R. Mitkov (Ed.), *The Oxford handbook of computational linguistics* (2nd ed, pp. 201-218). Oxford: Oxford University Press.

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Horsmann, T., Erbs, N., & Zesch, T. (2015). Fast or Accurate? – A Comparative Evaluation of PoS Tagging Models. In *Proceedings of the Int. Conference of the German Society for*

Computational Linguistics and Language Technology (pp. 22-30). University of Duisburg-Essen, Germany.

International Organization for Standardization / International Electrotechnical Commission. (2011). ISO/IEC 25010:2011 (E) Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva. [Retrieved July 2021 from https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en]

Jijo, B. T., & Mohsin Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. In *Journal of Applied Science and Technology Trends* (Vol. 2, pp. 20-28).

Jurafsky, D., & Martin H. J. (2020). Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Draft of December 30, 2020. [Retrieved from https://web.stanford.edu/~jurafsky/slp3/]

Keung, P., Lu, Y., Szarvas, G., & Smith, N. A. (2020). The Multilingual Amazon Reviews Corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Leech, G., & Wilson, A. (1996). EAGLES recommendations for the morphosyntactic annotation of corpora. Technical Report EAG-TCWG-MAC/R, ILC-CNR, Pisa.

Leech, G., & Smith, N. (1999). The Use of Tagging. In H. van Halteren (Ed.), *Syntactic Wordclass Tagging* (Text Speech and Language Technology, Vol. 9, pp. 23-36). Dordrecht, Boston and London: Kluwer Academic Publishers.

Magistry, P., Ligozat, A. L., & Rosset, S. (2019). Exploiting languages proximity for part-of-speech tagging of three French regional languages. In *Language Resources and Evaluation* (53(4), pp. 865-888) https://doi.org/10.1007/s10579-019-09463-7

Meyer, C. F. (2002). Annotating a corpus. In *English Corpus Linguistics: An Introduction* (Studies in English Language, pp. 81-99). Cambridge: Cambridge University Press.

Mikheev, A. (2004). Text segmentation. In R. Mitkov (Ed.), *The Oxford handbook of computational linguistics* (2nd ed, pp. 201-218). Oxford: Oxford University Press.

Morton, T., Kottmann, J., Baldridge, J., & Bierner, G. (2005). Opennlp: A java-based nlp toolkit.

Müller, T., Schmid, H. & Schütze, H. (2013). Efficient Higher-Order {CRF}s for Morphological Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 322-332). Association for Computational Linguistics. USA.

Nivre, J., de Marneffe, M., Ginter, F., Goldberg, Y., Jan Hajič, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., & Zeman, D. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of LREC*.

Nooralahzadeh, F., Brun, C., & Roux, C. (2014). Part of Speech Tagging for French Social Media Data. In *COLING 2014, 25th International Conference on Computational Linguistics* (Proceedings of the Conference: Technical Papers, pp. 81-99). Dublin, Ireland.

Padró, L. (1998) A Hybrid Environment for Syntax-Semantic Tagging. (PhD thesis, Dept. Llenguatges i Sistemes Informàtics). Universitat Politècnica de Catalunya.

Paroubek, P. (2000). Language Resources as by-Product of Evaluation: The MULTITAG Example. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00).* Athens, Greece: European Language Resources Association (ELRA).

Paroubek, P. (2007). Evaluating Part-of-Speech Tagging and Parsing. In L. Dybkjær , H. Hemsen, W. Minker (Eds.), *Evaluation of Text and Speech Systems*
(Text, Speech and Language Technology, pp. 99-124). Springer.

Paroubek, P., Chaudiron, S., Hirschman, L. (2007). Principles of Evaluation in Natural Language Processing. In *Revue TAL* (Vol. 48(1), pp. 7-31). ATALA (Association pour le Traitement Automatique des Langues).

Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-Of-Speech Tagging. EMNLP.

Ron, A. (2017). Inter-annotator Agreement. In N. Ide, J. Pustejovsky (Eds.), *Handbook of Linguistic Annotation* (pp. 297-313). Springer Netherlands.

Sagot, B. (2010). The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10).* Valletta, Malta: European Language Resources Association (ELRA).

Savary, A., Cordeiro, S. R., & Ramisch, C. (2019). Without lexicons, multiword expression identification will never fly: A position statement. Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019). Florence, Italy.

Schachter, P. (1985). Part-of-speech systems. In T. Shopen (Ed.), *Language Typology and Syntactic Description: Clause Structure* (2nd ed, Vol. 1, pp. 1-10). Cambridge University Press.

Schmid, H. (1994a). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK.

Schmid, H. (1994b). Part-of-Speech Tagging with neural networks. In *COLING 1994, Proceedings of the International Conference on Computational Linguistics.*

Schmid, H. (2019). Deep Learning-Based Morphological Taggers and Lemmatizers for Annotating Historical Texts. In *Proceedings of Digital Access to Cultural Heritage* (pp. 133-137). Association for Computing Machinery, New York.

Schmidt, R. M. (2019). Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. ArXiv, abs/1912.05911.

Seuren, P. (2009). The clitics mechanism in French and Italian. In *Probus* (Vol. 21(1), pp. 83-142). https://doi.org/10.1515/prbs.2009.004

Simkó, K., Kovács, V., & Vincze, V. (2017). USzeged: Identifying Verbal Multiword Expressions with POS Tagging and Parsing Techniques. In *Proceedings of the 13th Workshop on Multiword Expressions* (MWE 2017) (pp. 48-53). Association for Computational Linguistics. Valencia, Spain.

Singh, S. (2018). Natural Language Processing for Information Extraction. ArXiv, abs/1807.02383.

Stein, A. (2003): Lexikalische Kookkurrenz im afrikanischen Französisch. Zeitschrift für französische Sprache und Literatur (Journal for French Language and Literature, Vol. 113, pp. 1-17).

Stosic, D., & Miletic, A. (2019). ParCoGLiJe [Corpus]. ORTOLANG (Open Resources and TOols for LANGuage) - www.ortolang.fr, v2, https://hdl.handle.net/11403/stosic/v2

Straka, M., & Straková, J. (2017). Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies.* Vancouver, Canada.

Straka, M. (2018). UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of CoNLL 2018: The SIGNLL Conference on Computational Natural Language Learning* (pp. 197-207). Association for Computational Linguistics. Stroudsburg, PA, USA.

Sutton, C., & McCallum, A. (2010). An Introduction to Conditional Random Fields. ArXiv, abs/1011.4088.

Toutanova, K., Klein, D., Manning, C.D. & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03).* (Vol. 1, pp. 173-180) Association for Computational Linguistics. USA. https://doi.org/10.3115/1073445.1073478

Variš, D., & Klyueva, N. (2018). Improving a Neural-based Tagger for Multiword Expressions Identification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA). Miyazaki, Japan.

Voutilainen, A. (1999). A Short History of Tagging. In H. van Halteren (Ed.), *Syntactic Wordclass Tagging* (Text Speech and Language Technology, Vol. 9, pp. 9-21). Dordrecht, Boston and London: Kluwer Academic Publishers.

Zinsmeister, H., Hinrichs, E., Kübler, S., & Witt, A. (2008). Linguistically annotated corpora: Quality assurance, reusability and sustainability. In A. Lüdeling & M. Kytö (Eds.), *Corpus linguistics: an international handbook.* (Vol.1, pp. 759-776). Berlin: Mouton de Gruyter.

Zeroual, I., & Lakhouaja, A. (2019). A Comparative Study of Standard Part-of-Speech Taggers.

8. Annexes

Annex 1 – Original corpora

TCOF (ATILF, 2020)

1st recording: general meeting of a pétanque club.

assemblee_sar_08.trs

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">

<Trans scribe="" audio_filename="assemblee_sar_08.wav" version="28" version_date="131019">
  <Speakers>
    <Speaker id="spk1" name="L1" check="no" type="male" dialect="native" accent="" scope="local" />
    <Speaker id="spk2" name="L2" check="no" type="female" dialect="native" accent="" scope="local" />
    <Speaker id="spk3" name="L3" check="no" type="female" dialect="native" accent="" scope="local" />
    <Speaker id="spk4" name="L4" check="no" type="female" dialect="native" accent="" scope="local" />
    <Speaker id="spk5" name="L5" check="no" dialect="native" accent="" scope="local" />
    <Speaker id="spk7" name="LX" check="no" dialect="native" accent="" scope="local" />
  </Speakers>
  <Episode>
    <Section type="report" startTime="0.0" endTime="959.171">
      <Turn speaker="spk1" mode="spontaneous" fidelity="high" startTime="0.0" endTime="213.562">
        <Sync time="0.0" />
        bon ben on va commencer l'assemblée générale alors
        <Sync time="2.634" />
        <Event desc="conversations de fond" type="noise" extent="begin" />
        <Sync time="3.76" />
        donc je déclare l'assemblée générale euh 2008 <Event desc="pause" type="noise" extent="instantaneous"/>
        <Sync time="8.814" />
        ouverte à 18h40 <Event desc="pause" type="noise" extent="instantaneous"/>
        <Sync time="12.264" />
        donc je souhaite la bienvenue ben <Event desc="pause" type="noise" extent="instantaneous"/> à Madame =N1= adjointe euh au sport à la ville de =T1= <Event desc=
        "pause" type="noise" extent="instantaneous"/>
        <Sync time="19.176" />
        à Monsieur =N2= président de l'O.M.S.
        <Comment desc="sigle=Office Municipal des Sports" />
        et à Monsieur =P1= =N3= président du comité en =T2= <Event desc="pause" type="noise" extent="instantaneous"/>
        <Sync time="25.718" />
        qui a été élu euh récemment euh <Event desc="pause" type="noise" extent="instantaneous"/> il y a une semaine <Event desc="pause" type="noise" extent="instantaneous"
        />
        <Sync time="29.892" />
        et ainsi qu'à tous les membres et <Event desc="pause" type="noise" extent="instantaneous"/> licenciés <Event desc="pause" type="noise" extent="instantaneous"/> du
        club <Event desc="pause" type="noise" extent="instantaneous"/>
        <Sync time="36.409" />
        donc avant de dib- débuter l'A.G.
        <Comment desc="sigle=Assemblée Générale " />
        ben je vais demander une minute euh de silence
        <Sync time="39.414" />
        en la mémoire de ceux qui ont eu <Event desc="pause" type="noise" extent="instantaneous"/> la douleur de perdre un membre de leur famille <Event desc="pause" type=
        "noise" extent="instantaneous"/>
        <Sync time="44.565" />
        mais aussi on aura une pensée particulière euh par rapport à Monsieur =N4= =P2= <Event desc="pause" type="noise" extent="instantaneous"/>
        <Sync time="51.233" />
        membre fondateur du comité =T2= et président de <Event desc="pause" type="noise" extent="instantaneous"/> de celui-ci pendant plus de seize ans
        <Sync time="56.59" />
        <Event desc="bruit de chaises" type="noise" extent="instantaneous" />
        <Sync time="61.88" />
        <Comment desc="pause longue" />
```

2nd recording: interview of a professional rock climber.

escalade_mic.trs

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">

<Trans scribe="iclement" audio_filename="escalade_mic.wav" version="15" version_date="110421">
  <Speakers>
    <Speaker id="spk1" name="L1" scope="local" />
    <Speaker id="spk2" name="L2" scope="local" />
  </Speakers>
  <Episode>
    <Section type="report" startTime="0.0" endTime="1387.017">
      <Turn speaker="spk1" startTime="0.0" endTime="7.751">
        <Sync time="0.0" />
        alors =P1= je je sais que tu fais de l'escalade
        <Sync time="2.218" />
        est-ce que tu pourrais m'en dire plus puisque c'est un sport que je connais pas trop bien et puis ça m'intéresse
      </Turn>
      <Turn speaker="spk2" startTime="7.751" endTime="26.118">
        <Sync time="7.751" />
        alors plutôt que de sport je parlerais d'activité escalade +
        <Sync time="10.79" />
        euh l'activité escalade en tant en tant que telle c'est une des activités
        <Sync time="15.874" />
        si on la compare à d'autres à d'autres qui est euh + la plus riche ou euh en tout cas très diversifiée
      </Turn>
      <Turn speaker="spk1" startTime="26.118" endTime="27.048">
        <Sync time="26.118" />
        pourquoi tu dis ça
      </Turn>
      <Turn speaker="spk2" startTime="27.048" endTime="42.453">
        <Sync time="27.048" />
        parce qu'elle se pratique euh aussi bien en nature que sur une structure artificielle dans un gymnase
        <Sync time="34.03" />
        que sur des montagnes de huit mille mètres de haut des falaises de quatre cents mètres
        <Sync time="37.332" />
        ou des blocs de trois mètres de haut +
        <Sync time="39.516" />
        avec cordes sans cordes avec échelle sans échelle euh
      </Turn>
      <Turn speaker="spk1" startTime="42.453" endTime="46.341">
        <Sync time="42.453" />
        et toi tu tu fais quoi plutôt euh + plutôt en salle ou
      </Turn>
      <Turn speaker="spk2" startTime="46.341" endTime="75.248">
        <Sync time="46.341" />
        moi je + s- me sers du support euh de la structure artificielle d'escalade envie euh d'aller euh m'éclater en falaise
        <Sync time="55.802" />
        donc euh falaise euh que ce soit une falaise dite sportive c'est-à-dire une falaise qui fait trente mètres d'esca-
        <Sync time="62.117" />
        trente mètres de +
        <Sync time="63.747" />
        euh donc des voies d'une lon- d'une seule longueur ou alors des voies qui font quatre cents mètres +
```

ParCoGLiJe (Stosic & Miletic, 2019)

It contains the following books and their translation:

(1) Daudet, Alphonse (1869), *Lettres de mon moulin;*

(2) Dumas, Alexandre (1844), *Les trois mousquetaires*;

(3) Comtesse de Ségur (1860), *Mémoires d'un âne;*

(4) Verne, Jules (1870), *Vingt mille lieues sous les mers;*

(5) Dickens, Charles (1837), *Oliver Twist;*

(6) Hodgson Burnett, Frances (1911), *The Secret Garden;*

(7) Kipling, Rudyard (1894), *Jungle book;*

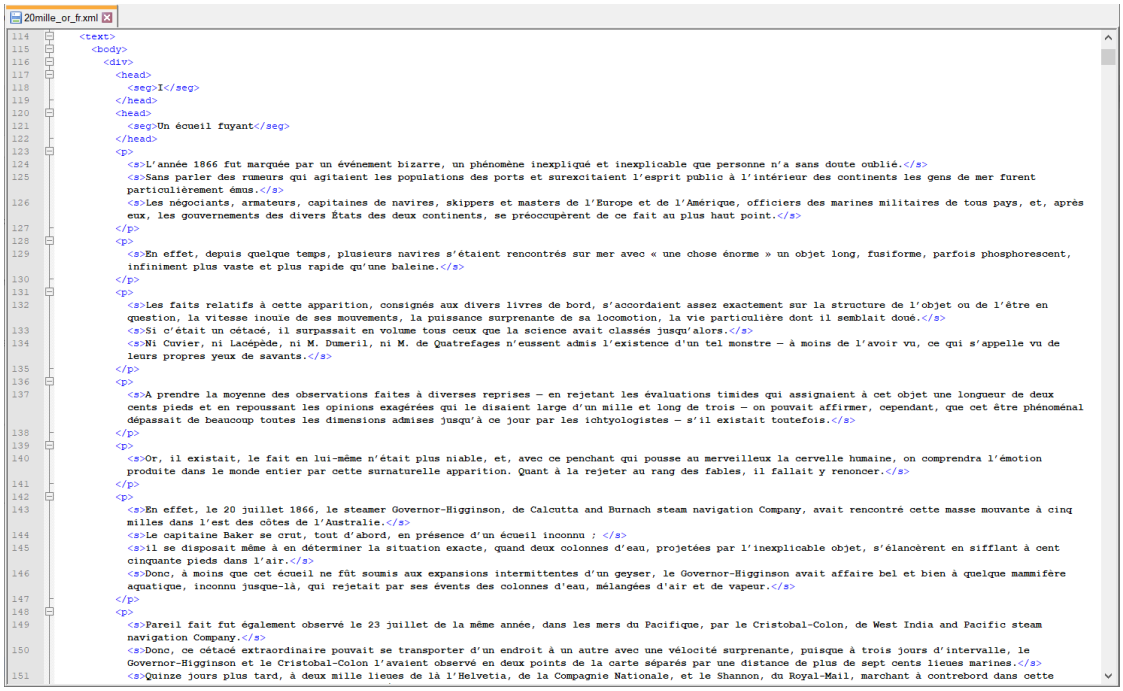
8) Stevenson, Robert Louis (1883), *Treasure Island;*

20000 Lieues sous les mers : Metadata in TEI-P5 standard

20mille_or_fr.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<teiCorpus>
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>20000 Lieues sous les mers</title>
        <subtitle>Parts 1 et 2</subtitle>
        <author>
          <name>
            <forename>Jules</forename>
            <surname>Verne</surname>
          </name>
        </author>
        <respStmt>
          <resp>translator</resp>
          <resp>transcriber</resp>
          <name>
            <forename/>
            <surname/>
          </name>
          <name/>
        </respStmt>
      </titleStmt>
      <editionStmt>
        <edition>No remarks</edition>
      </editionStmt>
      <extent>
        <measure unit="words" quantity="142959"/>
      </extent>
      <publicationStmt>
        <publisher>J. HETZEL ET Cie, 18 rue Jacob, Paris</publisher>
        <pubPlace>Paris</pubPlace>
        <date>2004</date>
        <availability>
          <licence>public</licence>
        </availability>
      </publicationStmt>
      <notesStmt>
        <note>This file was created by the ParCoLab team as part of the ParCoGliJe project financed by the CORLI consortium of the TGIR HumaNum, the ORTOLANG Equipex, the
        COCOON platform and the Audio Document Service of the BNF.</note>
      </notesStmt>
      <sourceDesc>
        <p>Downloaded from http://www.gutenberg.org/files/5097/5097-h.htm</p>
      </sourceDesc>
    </fileDesc>
    <profileDesc>
      <langUsage>
        <language ident="fr" langOri="fr"/>
      </langUsage>
      <creation>
        <date>1869</date>
```

20000 Lieues sous les mers : Text



The Multilingual Amazon Reviews Corpus

UNAVAILABLE

CHEU-lex

A parallel and comparable trilingual corpus (de, fr, it) containing in total 792 texts (444 bilateral agreements between Switzerland and EU plus 348 Swiss laws and ordinances).

cheu-lex_fr.xml

```
<corpus>
<text date_entry="1 août 1971" date_signature="24 mars 1972" date_status="NA" decade_entry="1970" id="0.192.122.974" original_text="Y" topic_macro="0.1 Droit international public général" topic_micro="0.19 Relations diplomatiques et consulaires. Missions spéciales. Organisations internationales. Règlements des conflits. Reconduction d'accords" type="agreement" url="https://www.admin.ch/opc/fr/classified-compilation/19720062/index.html">
<title>
<s id="agr000_fr_1">Accord sur le statut juridique en Suisse de la Banque Européenne d'Investissement</s>
</title>
<title_info>
<s id="agr000_fr_2">Conclu le 24 mars 1972 Approuvé par l'Assemblée fédérale le 27 septembre 1972 Instrument de ratification suisse notifié le 14 novembre 1972 Entré en vigueur avec effet dès le 1er août 1971
(Etat le 1er juillet 1974)</s>
</title_info>
<preamble>
<s id="agr000_fr_3">Le Conseil fédéral suisse et la Banque Européenne d'Investissement
avec siège provisoire à Luxembourg
ont conclu l'accord ci-après en vue de déterminer le statut juridique en Suisse de la Banque et de ses fonctionnaires.</s>
</preamble>
<body>
<article_title>
<s id="agr000_fr_4">Titre I Personnalité juridique
Art. 1</s>
</article_title>
<article_text>
<s id="agr000_fr_5">Le Conseil fédéral reconnaît la personnalité internationale et la capacité juridique de la Banque Européenne d'Investissement.</s>
</article_text>
<article_title>
<s id="agr000_fr_6">Titre II Juridiction des tribunaux suisses
Art. 2</s>
</article_title>
<article_text>
<s id="agr000_fr_7">La Banque ne peut être assujettie à la juridiction des tribunaux suisses que dans les cas et sous les modalités prévus dans le présent titre.</s>
</article_text>
<article_title>
<s id="agr000_fr_8">Art. 3</s>
</article_title>
<article_text>
<s id="agr000_fr_9">La Banque peut être actionnée devant les tribunaux suisses pourvu qu'elle (i) ait établi un bureau en Suisse; ou (ii) ait élu domicile en Suisse aux fins de recevoir des significations ou notifications de sommations; ou (iii) ait émis ou garanti des titres en Suisse.</s>
</article_text>
<article_title>
<s id="agr000_fr_10">Art. 4</s>
</article_title>
<article_text>
<s id="agr000_fr_11">La Banque ne pourra cependant pas être actionnée par des Etats membres ou par des personnes agissant pour le compte desdits membres ou tirant leurs droits de ceux-ci.</s>
</article_text>
<article_title>
<s id="agr000_fr_12">Art. 5</s>
</article_title>
<article_text>
<s id="agr000_fr_13">Les biens et avoirs de la Banque ne pourront faire l'objet d'aucune forme de saisie ou exécution tant qu'un jugement passé en force de chose
```

cheu-lex_fr.xml

```
183904 <text date_entry="NA" date_signature="17 décembre 2004" date_status="NA" decade_entry="NA" id="NA" original_text="NA" topic_macro="0.6 Finances" topic_micro="NA" type="law" url="https://www.admin.ch/opc/fr/official-compilation/2005/2185.pdf">
183905 <title>
183906 <s id="ord114_fr_1">Arrêté fédéral
183907 portant approbation de l'accord bilatéral entre la Suisse et les Communautés européennes sur l'imposition des fonctionnaires européens retraités résidant en Suisse</s>
183908 </title>
183909 <title_info>
183910 <s id="ord114_fr_2">du 17 décembre 2004</s>
183911 </title_info>
183912 <preamble>
183913 <s id="ord114_fr_3">L'Assemblée fédérale de la Confédération suisse,
183914 vu les art. 54, al. 1, et 166, al. 2, de la Constitution,
183915 vu le message du Conseil fédéral du 1er octobre 2004,
183916 arrête:</s>
183917 </preamble>
183918 <body>
183919 <article_title>
183920 <s id="ord114_fr_4">Art. 1</s>
183921 </article_title>
183922 <article_text>
183923 <s id="ord114_fr_5">1 L'Accord du 26 octobre 2004 entre le Conseil fédéral suisse et la Commission des Communautés européennes en vue d'éviter la double imposition des fonctionnaires retraités des institutions et agences des Communautés européennes résidant en Suisse est approuvé.</s>
183924 <s id="ord114_fr_6">2 Le Conseil fédéral est autorisé à le ratifier.</s>
183925 </article_text>
183926 <article_title>
183927 <s id="ord114_fr_7">Art. 2</s>
183928 </article_title>
183929 <article_text>
183930 <s id="ord114_fr_8">Le présent arrêté est sujet au référendum prévu par l'art. 141, al. 1, let. d, ch. 3, Cst. pour les traités internationaux qui contiennent des dispositions importantes fixant des règles de droit ou dont la mise en œuvre exige l'adoption de lois fédérales.</s>
183931 <s id="ord114_fr_9">Conseil des Etats, 17 décembre 2004
183932 Le président: Bruno Frick
183933 Le secrétaire: Christoph Lanz</s>
183934 <s id="ord114_fr_10">Conseil national, 17 décembre 2004
183935 Le président: Jean-Philippe Maitre
183936 Le secrétaire: Christophe Thomann</s>
183937 <s id="ord114_fr_11">Expiration du délai référendaire</s>
183938 <s id="ord114_fr_12">Le délai référendaire s'appliquant au présent arrêté a expiré le 31 mars 2005 sans avoir été utilisé.</s>
183939 <s id="ord114_fr_13">1er avril 2005 Chancellerie fédérale</s>
183940 </article_text>
183941 </body>
183942 </text>
183943 <text date_entry="NA" date_signature="12 juin 2009 " date_status="NA" decade_entry="NA" id="NA" original_text="NA" topic_macro="0.7 Travaux publics - Energie - Transports et communications" topic_micro="0.78 Postes et télécommunications" type="law" url="https://www.admin.ch/opc/fr/official-compilation/2010/3497.pdf">
183944 <title>
183945 <s id="ord115_fr_1">Arrêté fédéral
183946 portant approbation de l'accord sur la participation suisse au programme communautaire MEDIA pour les années 2007 à 2013</s>
183947 </title>
183948 <title_info>
183949 <s id="ord115_fr_2">du 12 juin 2009</s>
183950 </title_info>
```

Annex 2 – Links to the Test corpora and the Gold-standard on GitHub

Test Corpora

Spoken

- TXT
- XML

Literature

- TXT
- XML

Review

UNAVAILABLE

Law

- TXT
- XML

Evaluation Corpus

- One-sentence-per-line layout (with line breaks between each sentence)
- One-token-per-line layout

Gold-standard Corpus

The Gold-standard corpus is a version of the Evaluation corpus annotated with POS tags, morphological features (which have not been checked for consistency) and lemmas. Only POS tags have been manually checked and corrected. For some tokens different POS tags alternatives are given: this is due to the fact that MElt and TreeTagger do not have the same tagset as UDPipe 2.0 which is the tagset chosen as reference.

Annex 3 – POS tagging systems

NAME	OpenNLP	Freeling	HunPos	MarMot
APPROACH	Probability model	(1) Trigram Markov model (2) Statistical with handwritten constraint	Second-order Markov model	Highr-order Conditional Random Field (CRF)
FRENCH MODEL	yes	yes	no	no
URL	https://opennlp.apache.org/	http://nlp.lsi.upc.edu/freeling/index.php	https://code.google.com/archive/p/hunpos/	http://cistern.cis.lmu.de/marmot/
PAPER	Morton et al., 2005	(1) (based on) Brants, 2000 (2) Padró, 1998	(based on) Brants, 2000	Müller et al., 2013
NAME	MElt	RNNtagger	SMVTool	spaCy
APPROACH	Maximum entropy Markov model (MEMMs)	Bidirectional long short-term memory networks (LSTMs) with attention	Generator of sequential taggers based on Support Vector Machines	Different convolutional neural network models
FRENCH MODEL	yes	yes	no	yes
URL	http://almanach.inria.fr/software_and_resources/custom/MElt-fr.html	https://cis.uni-muenchen.de/~schmid/tools/RNNTagger/	https://www.cs.upc.edu/~nlp/SVMTool/#	https://spacy.io/models/fr
PAPER	Denis and Sagot, 2012	Schmid, 2019	Giménez & Márquez, 2004	Honnibal & Montani, 2017
NAME	Stanford	Tnt	TreeTagger	UDPipe 2.0
APPROACH	Maximum entropy model with a Cyclic Dependency Network	Implementation of the Viterbi algorithm for a second-order Markov model	Probabilistic model with decision trees	Artificial neural network with a single joint model
FRENCH MODEL	yes	no	yes	yes
URL	https://nlp.stanford.edu/software/tagger.html	https://www.coli.uni-saarland.de/~thorsten/tnt/	https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/	https://lindat.mff.cuni.cz/services/udpipe/
PAPER	Toutanova et al., 2003	Brants, 2000	Schmid, 1994a	Straka & Straková, 2017

More POS taggers are available at https://www.clarin.eu/resource-families/tools-part-speech-tagging-and-lemmatization.

Annex 4 – Example of annotation of MElt

```
L'/DET/le
audacieux/ADJ/audacieux
Cyrus/NPP/Cyrus
Field/ET/*Field
,/PONCT/,
le/DET/le
promoteur/NC/promoteur
de/P/de
l'/DET/le
entreprise/NC/entreprise
,/PONCT/,
qui/PROREL/qui
y/CLO/cld|cll
risquait/V/risquer
toute/ADJ/tout
sa/DET/son
fortune/NC/fortune
,/PONCT/,
provoqua/V/provoquer
une/DET/un
nouvelle/ADJ/nouveau
souscription/NC/souscription
./PONCT/.

Le/DET/le
faisceau/NC/faisceau
de/P/de
fils/NC/fil|fils
conducteurs/NC/conducteur
isolés/VPP/isoler
```

dans/P/dans

une/DET/un

enveloppe/NC/enveloppe

de/P/de

gutta-percha/NC/gutta-percha

,/PONCT/,

était/V/être

protégé/VPP/protéger

par/P/par

un/DET/un

matelas/NC/matelas

de/P/de

matières/NC/matière

textiles/NC/textile

contenu/VPP/contenir

dans/P/dans

une/DET/un

armature/NC/armature

métallique/ADJ/métallique

./PONCT/.

Le/DET/le

Great-Eastern/NPP/*Great-Eastern

reprit/V/reprendre

la/DET/le

mer/NC/mer

le/DET/le

13/DET/*13

juillet/NC/juillet

1866/NC/*1866

./PONCT/.

[…]

Annex 5 – Example of annotation of TreeTagger

```
«	PUN:cit	«
C'	PRO:DEM	ce
est	VER:pres	être
ici	ADV	ici
!	SENT	!
»	PUN:cit	»

Je	PRO:PER	je
regardait	VER:impf	regarder
à	PRP	à
bâbord	NOM	bâbord
et	KON	et
je	PRO:PER	je
ne	ADV	ne
vis	VER:pres	vivre
rien	ADV	rien
que	KON	que
l'	DET:ART	le
immensité	NOM	immensité
des	PRP:det	du
eaux	NOM	eau
tranquilles	ADJ	tranquille
.	SENT	.

On	PRO:PER	on
eût	VER:subi	avoir
dit	VER:pper	dire
des	PRP:det	du
ruines	NOM	ruine
ensevelies	VER:pper	ensevelir
```

sous PRP sous

un DET:ART un

empâtement NOM empâtement

de PRP de

coquilles NOM coquille

blanchâtres ADJ blanchâtre

commeKON comme

sous PRP sous

un DET:ART un

manteau NOM manteau

de PRP de

neige NOM neige

. SENT .

En PRP en

examinant VER:ppre examiner

attentivement ADV attentivement

cette PRO:DEM ce

masse NOM masse

, PUN ,

je PRO:PER je

crus VER:simp croire

reconnaître VER:infi reconnaître

les DET:ART le

formes NOM forme

épaissies VER:pper épaissir

d' PRP de

un DET:ART un

navire NOM navire

, PUN ,

rasé VER:pper raser

[…]

Annex 6 – Example of annotation of UDPipe

```
# text = en règle générale il y a beaucoup de fissures donc ils ont inventé un système avec des cames et un et un piston
1	en	en	ADP	_	_
2	règle	règle	NOUN	_	Gender=Fem|Number=Sing
3	générale	général	ADJ	_	Gender=Fem|Number=Sing
4	il	il	PRON	_	Gender=Masc|Number=Sing|Person=3|PronType=Prs
5	y	y	PRON	_	_
6	a	avoir	VERB	_	Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin
7	beaucoup	beaucoup	ADV	_	_
8	de	de	ADP	_	_
9	fissures	fissure	NOUN	_	Gender=Fem|Number=Plur
10	donc	donc	CCONJ	_	_
11	ils	il	PRON	_	Gender=Masc|Number=Plur|Person=3|PronType=Prs
12	ont	avoir	AUX	_	Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin
13	inventé	inventer	VERB	_	Gender=Masc|Number=Sing|Tense=Past|VerbForm=Part
14	un	un	DET	_	Definite=Ind|Gender=Masc|Number=Sing|PronType=Art
15	système	système	NOUN	_	Gender=Masc|Number=Sing
16	avec	avec	ADP	_	_
17-18	des	_	_	_	_	_	_	_
17	de	de	ADP	_	_
18	les	le	DET	_	Definite=Def|Gender=Fem|Number=Plur|PronType=Art
19	cames	came	NOUN	_	Gender=Fem|Number=Plur
20	et	et	CCONJ	_	_
21	un	un	DET	_	_
22	et	et	CCONJ	_	_
```

```
23      un      un      DET     _       _
24      piston  piston  NOUN    _       Gender=Masc|Number=Sing

# text = quand on actionne ce piston les cames se resserrent on le met dans la fissure on relâche le piston et les cames s'écartent
1       quand   quand   SCONJ   _       _
2       on      on      PRON    _       Gender=Masc|Number=Sing|Person=3
3       actionne        actionner       VERB    _
        Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin
4       ce      ce      DET     _       Gender=Masc|Number=Sing|PronType=Dem
5       piston  piston  NOUN    _       Gender=Masc|Number=Sing
6       les     le      DET     _
        Definite=Def|Gender=Fem|Number=Plur|PronType=Art
7       cames   came    NOUN    _       Gender=Fem|Number=Plur
8       se      se      PRON    _       Person=3|PronType=Prs
9       resserrent      resserrer       VERB    _
        Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin
10      on      on      PRON    _       Gender=Masc|Number=Sing|Person=3
11      le      le      PRON    _       Gender=Masc|Number=Sing|Person=3|PronType=Prs
12      met     mettre  VERB    _
        Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin
13      dans    dans    ADP     _       _
14      la      le      DET     _
        Definite=Def|Gender=Fem|Number=Sing|PronType=Art
15      fissure fissure NOUN    _       Gender=Fem|Number=Sing
16      on      on      PRON    _       Gender=Masc|Number=Sing|Person=3
17      relâcherelâcher         VERB    _
        Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin

18      le      le      DET     _
        Definite=Def|Gender=Masc|Number=Sing|PronType=Art
                                        […]
```