
# UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS


École Doctorale MIPTIS

Laboratoire d'Informatique, équipe BdTln

## THÈSE présentée par :

**Damien NOUVEL**

Soutenue le : 20 novembre 2012

Pour obtenir le grade de : **Docteur de l'université François Rabelais de Tours**

Discipline / Spécialité : **Informatique**

# RECONNAISSANCE DES ENTITÉS NOMMÉES PAR EXPLORATION DE RÈGLES D'ANNOTATION

## Interpréter les marqueurs d'annotation comme instructions de structuration locale

**THÈSE dirigée par :**

| | |
|---|---|
| M. Jean-Yves Antoine | Professeur des universités, Université François Rabelais de Tours |
| Mme Nathalie Friburger | Maître de conférences, Université François Rabelais de Tours |
| M. Arnaud Soulet | Maître de conférences, Université François Rabelais de Tours |

**JURY composé de :**

| | |
|---|---|
| M. Jean-Yves Antoine | Professeur des universités, Université François Rabelais de Tours |
| M. Frédéric Béchet | Professeur des universités, Aix Marseille Université |
| M. Bruno Crémilleux | Professeur des universités, Université de Caen (rapporteur) |
| Mme Nathalie Friburger | Maître de conférences, Université François Rabelais de Tours |
| Mme Sophie Rosset | Directrice de recherche, LIMSI-CNRS (rapporteure) |
| M. Arnaud Soulet | Maître de conférences, Université François Rabelais de Tours |

# Remerciements

Je voudrais remercier Sophie Rosset, Bruno Crémilleux et Frédéric Béchet, pour la richesse des échanges que nous avons eus au cours de ma thèse, pour avoir accepté d'examiner ce manuscrit et pour m'avoir fait part de leurs avis. Leurs réflexions ont beaucoup alimenté les miennes et m'ont beaucoup apporté.

Merci à Jean-Yves Antoine, Nathalie Friburger et Arnaud Soulet pour m'avoir accordé leur confiance. Un savant compromis d'encadrement et de liberté nous a permis d'être la plupart du temps en accord sur les directions à prendre. Et dans les autres cas, d'avoir des discussions passionnantes. Entre autres apprentissages, je loue en particulier leur recul, leur intuition et leur rigueur scientifique.

Ces quelques années au sein de l'équipe BDTLN, ont été l'occasion de rencontres et d'échanges quotidiens, toujours dans la bonne humeur, notamment avec Denis Maurel, Agata Savary, Patrick Marcel, Arnaud Giacometti, Thomas Devogele, Béatrice Bouchou-Markov, Verónika Peralta, Yacine Sam, Haoyuan Li, Nizar Messai.

J'ai eu la chance, avant et pendant mon doctorat, d'apprendre beaucoup, dans mon domaine et dans des domaines connexes. Je dois ceci à de trop nombreuses personnes, dont dernièrement Gaël de Chalendar, Olivier Ferret, Romaric Besançon, Pierre Zweigenbaum, Patrick Paroubek, Olivier Galibert, Maud Ehrmann, Christian Raymond, Eric de la Clergerie, Benoît Sagot, Rosa Stern, Isabelle Tellier, Thierry Charnois, Peggy Cellier...

Beaucoup de bonnes choses et au plaisir de se revoir à quelques camarades de route : Elsa Nègre, Marie Ndiaye, Harinaina Ravelomanantsoa, Cheikh Niang, Julien Aligon, Mouhamadou Saliou Diallo, Romain Lalande, Nadia Okinina. Mercis à Thierry Ressault, Aurore Leroy et Christelle Grange pour tout ce qu'ils ont facilité.

Merci en particulier à Pierre-François Laurand : remarquablement réactif et efficace, même les soirs et les week-ends ! Bon repos à vega et iup0217, instruments rudement sollicités dans le labeur. Et quelques pensées pour radium, vaillante passerelle, vieille garde qui m'a maintes fois salué lors de mes visites distantes et nocturnes.

Le soutien, la confiance, les encouragements m'ont toujours été apportés par mes proches : grands mercis à mes parents, sœurs, frère, amis, cousins et cousines. Seul regret : ne pas avoir partagé avec vous toutes les profondeurs que recèlent les entités nommées... mais je ne sais si l'inverse est vrai, et je m'en tiendrai donc là afin que mes activités professionnelles vous paraissent toujours un peu "mystiques".

Ce travail n'aurait vu le jour sans Katty, toujours présente, indispensable pour moi, et Iris, petite fleur qui pousse et qui, à deux ans, défie déjà toutes les thèses du monde.

# Résumé

Ces dernières décennies, le développement considérable des technologies de l'information et de la communication a modifié en profondeur la manière dont nous avons accès aux connaissances. Face à l'afflux de données et à leur diversité, il est nécessaire de mettre au point des technologies performantes et robustes pour y rechercher des informations. Les entités nommées (personnes, lieux, organisations, dates, expressions numériques, marques, fonctions, etc.) sont sollicitées afin de catégoriser, indexer ou, plus généralement, manipuler des contenus. Notre travail porte sur leur reconnaissance et leur annotation au sein de transcriptions d'émissions radiodiffusées ou télévisuelles, dans le cadre des campagnes d'évaluation Ester2 et Etape. En première partie, nous abordons la problématique de la reconnaissance automatique des entités nommées. Nous y décrivons les analyses généralement conduites pour traiter le langage naturel, discutons diverses considérations à propos des entités nommées (rétrospective des notions couvertes, typologies, évaluation et annotation) et faisons un état de l'art des approches automatiques pour les reconnaître. A travers la caractérisation de leur nature linguistique et l'interprétation de l'annotation comme structuration locale, nous proposons une approche par instructions, fondée sur les marqueurs (balises) d'annotation, dont l'originalité consiste à considérer ces éléments isolément (début ou fin d'une annotation). En seconde partie, nous faisons état des travaux en fouille de données dont nous nous inspirons et présentons un cadre formel pour explorer les données. Les énoncés sont représentés comme séquences d'items enrichies (morpho-syntaxe, lexiques), tout en préservant les ambigüités à ce stade. Nous proposons une formulation alternative par segments, qui permet de limiter la combinatoire lors de l'exploration. Les motifs corrélés à un ou plusieurs marqueurs d'annotation sont extraits comme règles d'annotation. Celles-ci peuvent alors être utilisées par des modèles afin d'annoter des textes. La dernière partie décrit le cadre expérimental, quelques spécificités de l'implémentation du système (mXS) et les résultats obtenus. Nous montrons l'intérêt d'extraire largement les règles d'annotation, même celles qui présentent une moindre confiance. Nous expérimentons les motifs de segments, qui donnent de bonnes performances lorsqu'il s'agit de structurer les données en profondeur. Plus généralement, nous fournissons des résultats chiffrés relatifs aux performances du système à divers point de vue et dans diverses configurations. Ils montrent que l'approche que nous proposons est compétitive et qu'elle ouvre des perspectives dans le cadre de l'observation des langues naturelles et de l'annotation automatique à l'aide de techniques de fouille de données.

**Mots clés :** Traitement automatique des langues, fouille de données, entités nommées, règles d'annotation.

# Abstract

Those latest decades, the development of information and communication technologies has substantially modified the way we access knowledge. Facing the volume and the diversity of data streams, working out robust and efficient technologies to retrieve information becomes a necessity. In this context, Named Entities (persons, locations, organizations, numerical expressions, brands, functions, etc.) may be required in order to categorize, index or, more generaly, manipulate contents. Our work focuses on their recognition and annotation inside radio and TV broadcasts transcripts, in the context of Ester2 and Etape evaluation campaigns. In the first part, we introduce our problematic, the automatic recognition of named entities. We describe the commonly conducted analysis to process natural language, question the linguistic properties of named entities (related notions, typologies, evaluation and annotation) and describe state-of-the-art approaches. From their linguistic nature and by interpreting annotation as a local structuring, we propose an instruction-driven approach, based on annotation markers (tags), which originality consists in considering those elements in isolation. In the second part, we present the formalism used to explore data and introduce our formal framework. Sentences are represented as sequences of enriched items (morpho-syntax, lexicon) that preserve ambiguity. We also propose an alternative representation by segments that allows to limit combinatorial search. Patterns correlated to annotation markers are extracted as annotation rules. Those may be used by models so as to actually annotate texts. The last part presents the experimental framework, the implemented system (mXS) and the obtained results. We show the interrest of widely extracting annotation rules, even those of low confidence. We experiment segment patterns, that give interresting performances for deeply structured data. More generaly, we give results relative to performances of the system from diverse points of view and in diverse configurations. They show that the proposed approach is competitive and that it opens up perspectives for natural language observation and automatic annotation using data mining.

**Keywords :** Natural Language Processing, Named Entities, Data Mining, Annotation Rules.

# Table des matières

# Liste des tableaux

# Table des figures

# Liste des Algorithmes

# Introduction

*"Totalité organisée faite d'éléments solidaires ne pouvant être définis que les uns par rapport aux autres en fonction de leur place dans cette totalité."* (Système) [de Saussure, 1916]

*"Les structures logico-mathématiques, en leur infinité, ne sont localisables ni dans les objets ni dans le sujet à son point d'origine. Il n'y a donc d'acceptable qu'un constructivisme, mais dont la lourde tâche est d'expliquer à la fois le mécanisme de formation des nouveautés et le caractère de nécessité logique qu'elles acquièrent en cours de route."* (J. Piaget) [Piaget et Chomsky, 1975]

## Contexte

Ces dernières décennies, le développement considérable des technologies de l'information et de la communication a modifié en profondeur la manière dont nous manipulons les connaissances. Outre l'accroissement indéniable des volumes de données mis à disposition, nous constatons une diversité toujours plus importante des types de contenus échangés. L'adoption massive des technologies numériques par les médias et les administrations nécessite de résoudre de nombreuses problématiques. Comment organiser et présenter les contenus ? Combien de temps faut-il mettre les documents à disposition du grand public et comment les archiver ? Quelles formats (texte, image, audio, vidéo) et quelles qualités sont adaptés aux utilisateurs et compatibles avec leurs équipements ? Et, pour ce qui nous concerne, comment y rechercher automatiquement des informations ?

Il s'agit donc de mettre au jour, au sein de ces contenus, des éléments qui permettront de les catégoriser, de les indexer ou, de manière plus générale, de les manipuler. Nous considérons cette problématique lorsque le contenu du document relève du langage naturel. En particulier, pour les contenus audio, le traitement automatique de la parole a été ces

dernières années l'objet de nombreux travaux de recherche. Les dernières avancées dans ce domaine permettent, sous certaines conditions, d'obtenir une transcription modérément bruitée du signal audio. Il devient alors envisageable de rechercher des informations au sein de textes, mais aussi de contenus audio (par exemple radiodiffusés) ou vidéo (par exemple télévisuels).

Or, lorsqu'il s'agit du langage naturel, les travaux en recherche d'information ont porté une attention particulière aux noms propres de personnes, de lieux et d'organisations, appelés *entités nommées*. Au gré des besoins, celles-ci ont été étendues aux dates, aux expressions numériques, aux marques ou aux fonctions, avant de recouvrir un large spectre d'expressions linguistiques. L'ensemble des expressions concernées paraissent *désigner* des entités du monde réel. Il est généralement accepté que les entités nommées sont des éléments plus particulièrement sollicités au sein des documents par les applications et, in fine, par l'utilisateur, dans un processus de recherche d'information.

## Motivations et contributions

En faisant des entités nommées notre objet d'étude, il s'agit en premier lieu de déterminer leur réalité linguistique. Si cet amalgame d'expressions linguistiques, apparemment très diverses, a émergé face à des besoins applicatifs nouveaux, il nous semble peu satisfaisant de s'en tenir à une définition qui évolue au gré de ces besoins. Nous constatons par ailleurs que les expressions concernées semblent désigner des objets (réels ou mentaux) et ne paraissent pas faire appel à des raisonnements complexes. Nous prendrons position sur leur nature linguistique et en proposerons une définition qui s'appuie sur deux propriétés en intention (*stabilité*, *opérabilité*). Pour leur traitement automatique, en tant qu'expressions du langage naturel, nous choisissons deux axes (*structurel*, *ontologique*) sur lesquels nous appuyer pour conduire les analyses.

Dans le cadre de programmes de recherche sur ces thématiques, des jeux de données ont été produits afin de caractériser les entités nommées ou de mesurer les performances des systèmes réalisant leur reconnaissance. En particulier, les *campagnes d'évaluation* comportent une phase d'*annotation* de données (corpus) qui permettent de mettre de systèmes en compétition. Au cours de cette phase, des directives sont édictées concernant les différentes catégories d'entités nommées à prendre en considération ainsi que la manière dont elles se structurent. Nous reprenons à notre compte cette procédure pour relever les entités nommées au sein de textes. En particulier, nous mettons au jour une mécanique reposant sur l'insertion de *marqueurs* (balises) d'annotation, interprétés comme instructions locales et sous-jacentes à la reconnaissance de structures d'entités nommées.

Pour élaborer un système reconnaissant automatiquement les entités nommées, nous faisons la distinction entre les approches *orientées connaissances*, qui reposent sur l'implémentation de procédures que l'on suppose efficaces pour la tâche considérée, et les approches *orientées données*, qui ajustent les paramètres de modèles numériques selon des exemples de la problématique et de sa solution. Notre positionnement nous permet de proposer une approche intermédiaire, qui considère l'insertion individuelle de marqueurs d'annotation au sein des données. En ceci, nous nous plaçons dans une perspective intermédiaire entre les systèmes orientés connaissances et les systèmes orientés données.

Les techniques de fouille de données nous permettent de formaliser notre approche comme l'exploration exhaustive et objective de données. Par suite, des motifs corréllés aux marqueurs peuvent être extraits, que nous appellons *règles d'annotation*, et que nous mettons à disposition de modèles numériques. Par ailleurs, nous proposons dans ce contexte une formulation alternative de *motifs de segments*, qui permettent de limiter la combinatoire lorsque l'on examine divers niveaux de granularité. Le système implémenté (mXS) est expérimenté et ses performance sont décrites pour diverses configuration, sur des jeux de données mis à disposition par les campagnes Ester2 et Etape.

## Organisation du manuscrit

La première partie aborde la problématique du traitement automatique du langage. Nous commençons par exposer la relation qu'entretient l'automate avec le langage et les diverses analyses généralement conduites pour traiter le langage naturel. Nous abordons ensuite la problématique des entités nommées, avec une rétrospective linguistique, par l'évocation de notions qu'elle recouvre, en décrivant les campagnes d'évaluation menées sur le sujet, puis en en proposant une définition en deux points. Nous concluons cette partie en faisant un état de l'art des approches automatiques pour reconnaître les entités nommées et par une proposition d'approche reposant sur la détection séparée des marqueurs qui débutent ou terminent les annotations.

En seconde partie, nous formalisons d'un point de vue théorique l'exploration de données pour la reconnaissance d'entités nommées. Un état de l'art de la fouille de données textuelles nous permet de situer notre approche. Nous décrivons ensuite les caractéristiques des motifs que nous explorons afin d'extraire des règles d'annotations à partir des données, notamment avec une formulation alternative par motifs de segments. Enfin, nous présentons les différentes déclinaisons de modèles que nous expérimentons lorsqu'il s'agit d'utiliser les règles extraites pour réaliser une annotation.

Dans la dernière partie, nous décrivons le système implémenté et les résultats obtenus. Nous détaillons en premier lieu les modules de traitements, ressources lexicales et corpus à notre disposition pour mener nos expériences. Nous donnons ensuite des informations détaillées sur la manière dont est implémentée l'exploration de données, dont sont extraites les règles d'annotation, et analysons les motifs obtenus. Enfin, nous décrivons les modèles numériques utilisés et fournissons des résultats chiffrés relatifs aux performances obtenues par le système, ainsi que des indicateurs supplémentaires quant à son comportement à divers points de vue et dans diverses configurations.

# Première partie

# Traitement automatique des langues et entités nommées

# Chapitre 1

# Traitement du langage par automates

## 1.1 Analyser automatiquement le langage naturel

Le travail que nous présentons dans ce manuscrit vise à élaborer des méthodes pour traiter le langage dit *naturel* de manière automatique. Plus particulièrement, notre objet d'étude est le langage oral en français (issue de transcriptions d'émissions radiodiffusées ou télévisuelles), dont la parole spontanée et les dialogues. Nous nous plaçons dès lors dans un contexte où la formulation d'idées est assez immédiate et, par conséquent, peu contrôlée, contrairement aux langages écrits ou artificiels (presse, romans, rapports, mathématiques, codes, langages de programmation, etc.). Les analyses que nous cherchons à réaliser ont pour objectif de construire des représentations stables sur lesquelles diverses applications puissent ensuite opérer selon leurs objectifs : extraction d'information, classification, indexation, résumé, etc.

Pour ce faire, nous aurons en premier lieu recours à des théories linguistiques qui sont issues, pour une large part, de l'observation du langage naturel. Cependant, si ces dernières parviennent à décrire de manière systématique certains *faits de langues*, nous chercherons en outre à implémenter des méthodes capables d'analyser *automatiquement* le langage naturel. Nous nous situons donc dans le contexte d'un paradigme plus récent, le *Traitement Automatique des Langues* (TAL), dont l'émergence est liée à l'apparition et à la montée en puissance des ordinateurs. L'étude et l'analyse du langage naturel dépend par suite des procédures qu'il est possible d'implémenter sous forme d'automate.

Dans ce chapitre, nous abordons quelques généralités concernant les approches qui se sont popularisées en TAL. L'objectif est de présenter les méthodes couramment utilisées pour traiter automatiquement le langage naturel et de nous positionner à leur égard. A cet effet, nous examinons la relation qu'entretient la théorie des automates avec les sciences du langage. Historiquement, les automates sont des machines adéquates pour exécuter un langage artificiel (programme), sans prétention à analyser le langage naturel. En conséquence, nous prendrons garde à ne pas considérer que les méthodes utilisées pour interpréter les langages artificiels sont implicitement adéquates pour traiter le langage naturel.

Mettre en place des procédures qui traitent le langage naturel, demande, pour une énonciation donnée, d'en distinguer les éléments constitutifs, d'expliciter leurs rôles et de

déterminer les interactions qui lient ces éléments, entre eux ou à une base de connaissances. De nombreuses approches ont été expérimentées à cet effet et ont conduit à l'émergence (entre autres) de méthodes dites *incrémentales* : diverses analyses sont menées, qui se superposent les unes aux autres. Chaque analyse considère le langage selon un certain degré de granularité : caractères, mots, expressions composées, syntagmes, propositions, énoncés, etc. Afin de donner une vision globale de leur utilisation en TAL, voici une description sommaires d'analyses couramment prises en compte lors de ce processus :

- **Tokenisation** : segmentation des données en unités signifiantes, les tokens (mots).
- **Lemmatisation** : normalisation des tokens afin de faire abstraction des variations de flexion (en grand part liées à la conjugaison et à la déclinaison).
- **Morpho-syntaxe** : affectation de catégories linguistiques (nom, verbe, déterminant, etc.) aux tokens selon le rôle qu'ils jouent au sein de l'énoncé.
- **Syntaxe** : détermination de relations syntaxiques (ou dépendances) liant des tokens ou des groupes de tokens entre eux (sujet, objet, complément, etc.).
- **Sémantique** : attribution de signification aux unités en présence afin de permettre des raisonnements, généralement d'ordre logique.

Nous remarquons que la tokenisation est une analyse également importante pour traiter les langages naturels ou artificiels. En revanche, il semble que la lemmatisation et la morpho-syntaxe revêtent plus d'importance pour les langages naturels. Inversement, les langages artificiels reposent beaucoup sur l'utilisation de règles syntaxiques, en particulier lors de la *programmation* d'automates. Pour nos besoins, il nous paraît pertinent de mesurer l'apport de ces diverses analyses (selon ce qui est observé au sein du langage naturel) et de faire abstraction, autant que possible, des méthodes dédiées à l'analyse des langages artificiels.

## 1.2 Des axes pour les analyses incrémentales ambiguës

Nous avons indiqué que les diverses analyses en jeu prennent part à un processus *incrémental*. Précisons cependant que ce mode d'analyse, dans le sens où nous l'entendons, implique que les analyses soient ordonnées et que les représentations obtenues prennent la forme d'arborescences. En revanche, cela n'empêche pas qu'une étape d'analyse particulière demeure indéterminée et émette plusieurs hypothèses. De fait, nous attachons une grande importance à la possibilité de manipuler des représentations potentiellement ambiguës, d'autant plus que des dépendances réciproques existent entre les divers modules de traitement. Considérons le fameux exemple :

*'La belle brise la glace'*

Sans contexte, cet énoncé présente une ambiguïté : il peut s'agir d'une *'belle'* (femme) qui *'brise la glace'*, ou bien d'une *'belle brise'* (fraîche), qui *'glace'* une personne. Ainsi, *'belle'* peut être un nom ou un adjectif, *'brise'* un verbe ou un nom, *'la'* un déterminant ou un pronom et *'glace'* un nom ou un verbe. Or des hypothèses peuvent être émises à partir de l'étape de lemmatisation (*'brise'* peut-être lemmatisé en *'brise'* le nom, ou *'briser'*, le verbe, de même pour *'glace'*). Mais la morpho-syntaxe puis la syntaxe contraignent les analyses de telle sorte que, sur l'ensemble de l'énoncé, seules deux hypothèses d'analyses sont vraisemblables en fin de processus pour cet énoncé. En outre, si nous disposons d'une information déterminante dans le co-texte (par exemple météorologique) nous pourrions

lever l'ambiguïté par une préférence d'ordre sémantique qui sélectionnerait rétroactivement une seule hypothèse en syntaxe, morpho-syntaxe et lemmatisation.

Or nous constatons que chaque niveau d'analyse semble s'appuyer à la fois sur la nature a priori de chaque unité (qui peut être partiellement déterminé par les analyses précédentes) et sur les contraintes issues de la contiguïté d'éléments au sein d'un énoncé. Idéalement, une analyse devrait être réalisée en prenant en considération toutes les unités de l'énoncé (ou éventuellement du document). Mais, à l'aide de traitements incrémentaux, l'objectif peut heureusement être simplifié en une *réduction* progressive de l'énoncé, au fur et à mesure que sont traitées les unités en présence et que se construit une représentation correspondante. Dans notre travail, nous considérons que ce mécanisme impose de mettre à disposition des analyses les deux axes suivants :

- **Ontologique** : une unité, de par sa nature, peut-être graduellement généralisée à une forme normale ou à un hyperonyme (subsomption, axe paradigmatique)
- **Structurel** : une analyse peut avoir à prendre en considération plusieurs unités contiguës (composition, axe syntagmatique)

Nous postulons que ces deux directions doivent être explorées de concert, même lorsque des ambiguïtés n'ont pas été résolues par des traitements préalables. Incrémentalement, chaque analyse utilise des motifs construits à partir du langage naturel partiellement traité en profondeur (*ontologie*) et en largeur (*structure*). L'utilisation de l'outil informatique doit nous permettre d'observer le langage naturel de manière aussi objective que possible : nous souhaitons faire moins appel à l'intuition linguistique qu'à l'observation sur corpus. Pour cela, nous mettrons à disposition de la machine ces deux axes d'analyse et chercherons à exhiber des motifs élaborés à l'aide de ces deux axes qui régissent certains phénomènes du langage naturel.

## 1.3 Principes généraux des automates et des grammaires

Le traitement automatique des langue vise à mettre en place des procédures informatiques qui permettent d'analyser le langage naturel. Dans ce contexte, la *machine de Turing* donne un aperçu du champ des possibles parmi les *programmes* dont est capable un automate. Signalons de prime abord que les données y sont représentées sous forme binaire (digitale) et par extension sous forme de *symboles*. Il est donc possible de projeter les unités signifiantes que l'on définit pour un langage (caractères, mots, phonèmes) vers l'*alphabet* que manipule la machine. In fine, le langage naturel est représenté sous la forme d'une longue séquence de symboles : la *bande* de la machine de Turing.

Cette machine réalise alors automatiquement des opérations sur la *séquence* de symboles donnée. Selon son état initial, les états et transitions définies par son programme, les symboles présents sur la bande, l'automate lit et modifie la bande jusqu'à atteindre un état final. Ainsi, les traitements, procédures ou raisonnements (généralisations, analogies, inférences, inductions) sont implémentées au sein des états et transitions de la machine.

Une formalisation communément admise comme socle théorique pour concilier automates et langages est la hiérarchie de Chomsky [Chomsky, 1956, Chomsky, 1957], dont les *grammaires* sont formulées à l'aide de *règles de production*. Outre les symboles appar-

tenant au vocabulaire en entrée $V$, un alphabet de symboles $N$, relatif à la grammaire implémentée, est manipulé sur la bande, en correspondance avec les analyses réalisées. Un mécanisme automatique permet alors de vérifier si un énoncé peut-être analysé à l'aide d'une grammaire donnée, et quel est le résultat de cette analyse. Nous donnons un récapitulatif succinct des formes que peuvent prendre les règles de production selon la hiérarchie des grammaires en table 1.1.

| **Type** | **Grammaire** | **Formalisme** | **Forme des règles de production** |
|---|---|---|---|
| 0 | Générale | Machine de Turing | $\alpha \to \beta \quad \alpha, \beta \in (V \cup N)^*$ |
| I | Contextuelle | Automates | $\alpha A \beta \to \alpha \gamma \beta \quad \alpha, \beta, \gamma \in V^*, A \in N$ |
| II | Hors-contexte | Automates à pile | $A \to \alpha \quad A \in N, \alpha \in (V \cup N)^*$ |
| III | Régulière | Automates à états finis | $A \to Ba, A \to a \quad A, B \in N, a \in V^*$ |

TABLE 1.1 – Hiérarchie de Chomsky

Au sein de cette hiérarchie, les approches en traitement du langage utilisent intensivement les grammaires de type II et III, qui présentent un bon compromis entre expressivité et complexité pour les besoins du TAL. Voici les applications qui en sont couramment faites :

- **Type III, automates à états finis** : reconnaissance des unités bas niveaux (tokenisation, morpho-syntaxe)
- **Type II, grammaires hors-contexte** : construction de structures (arbres syntaxiques et syntagmatiques, graphes de dépendances)

Si ces outils ont été élaborés pour analyser le langage naturel, nous remarquons qu'ils ont pour une large part été popularisés grâce à leur capacité à spécifier et analyser des langages artificiels. Effectivement, ces derniers sont aujourd'hui couramment formulés comme un ensemble de règles syntaxiques. Et pour cause : lors de l'interprétation d'un programme, son analyse doit être déterministe et ne pas contenir d'ambiguïtés avant son exécution. Les grammaires de type II et III sont adéquates à cet effet et permettent ainsi de programmer efficacement et précisément un automate par utilisation d'un langage dédié.

Mais en TAL, le constat est plus contrasté. Comme nous le détaillerons en section 3.1, les automates à états finis et grammaires hors-contexte sont aujourd'hui très répandus. Cependant, pour réaliser des analyses incrémentales, ces technologies nécessitent de prendre en compte l'intégralité des symboles des énoncés. Dans notre travail, nous interrogeons ce principe, qui nous paraît peu flexible face à la variabilité du langage naturel, en particulier dans le cadre des grammaires hors-contexte. Nous envisageons la possibilité, pour un humain ou pour une machine, de faire émerger le sens d'un énoncé sans exiger que chacun de ses éléments soient reconnus. En conséquence, nous proposons en section 3.3 un système de marqueurs, qui permet de déterminer séparément le début ou la fin d'une unité linguistique, sans nécessairement avoir à en analyser tous les constituants.

Le formalisme que nous expérimentons n'empêche cependant pas d'exploiter le résultats d'analyses préalables réalisées à l'aide d'automates ou de grammaires. Nous considérons celles-ci comme des *ressources*, dont en particulier :

- **Lexiques** (automates à états finis) : listes de tokens, lemmes, expressions composées
- **Transducteurs** (grammaires hors-contexte) : expressions linguistiques complexes

Ces ressources nous permettent d'obtenir des analyses (potentiellement ambiguës) pour la tokenisation, la lemmatisation, la morpho-syntaxe et la sémantique. Signalons dès à présent que nous ne présenterons pas de résultats utilisant des analyses syntaxiques. Ceci est dû à la difficulté de mettre en œuvre puis d'exploiter une telle analyse face au volume et à la complexité des données que nous manipulons (langage oral). Les quelques expériences préliminaires que nous avons faites n'ont pas été assez abouties pour pouvoir être présentées ici.

## 1.4 Degrés d'analyses de phénomènes linguistiques

Afin d'éviter toute ambiguïté sur les termes que nous emploierons au travers des discussions, nous précisons notre vocabulaire quant aux tâches à résoudre. Comme nous l'avons expliqué, nous cherchons à réaliser des analyses du langage naturel. Ces analyses, liées à la présence de phénomènes linguistiques, sont considérées comme des *problématiques* qui demandent à être *résolues* par un humain ou par une machine. Nous répartissons, au travers de nos considérations théoriques et expérimentales, l'analyse d'un phénomène observable particulier en sous-problématiques selon les informations qui sont collectées à son sujet :

- **Détection** : existence du phénomène donné (décision binaire)
- **Reconnaissance** : catégorisation du phénomène linguistique (parmi ses hyperonymes)
- **Résolution** : détermination de toutes les propriétés liées au phénomène

Cette division de la tâche vise à mieux discerner dans quelle mesure nous sommes capable d'analyser un phénomène donné, et d'évaluer le succès des analyses réalisées selon cette grille. Dans le contexte des *entités nommées* (pour simplifier, assimilons-les ici aux *noms propres*) et pour une énonciation donnée, les *détecter* consiste à déterminer où elles se trouvent (sans les typer), les *reconnaître* demande de surcroît à leur attribuer un type (ou une classe, une catégorie), les *résoudre* consiste à faire le lien entre l'entité nommée et un objet du monde réel (ou du discours, du référentiel) dont les propriétés sont supposées connues. Nous notons qu'il est possible d'utiliser la même classification lorsqu'il s'agit d'analyser les coréférences au sein du langage naturel, que l'on cherche également à détecter, à reconnaître et à résoudre.

# Chapitre 2

# Les Entités Nommées

## 2.1 Introduction

Les méthodes que nous présentons ici visent à reconnaître automatiquement les *"entités nommées"* au sein du langage naturel. De prime abord, mentionnons que ces entités nommées sont de fait associées à des expressions linguistiques sollicitées par des applications qui manipulent des documents textuels. Initialement, ce besoin s'est matérialisé par la nécessité de détecter les noms propres au sein de textes et, plus particulièrement, de reconnaître les *personnes*, les *lieux* et les *organisations*, éventuellement de les résoudre, afin de déterminer de *quoi* parle un texte.

Rapidement, d'autres objets se sont avérés utiles afin de mieux extraire et rechercher des informations au sein des documents, dont les *expressions de temps* (dates) et les *quantités monétaires* (montants). Il s'agit de construire des représentations élaborées du contenu de documents textuels, grâce à la reconnaissance d'expressions linguistiques complexes en leur sein. De fil en aiguille, de nombreux autres types d'expressions s'y sont ajoutés : marques, fonctions, bâtiments, produits, œuvres, lois, événements, etc. Au point qu'aujourd'hui, les entités nommées paraissent correspondre à un amalgame d'expressions très diverses, dont l'utilité pour des visées applicatives est établie, qui sont généralement décrites extensivement, mais pour lesquelles il existe bien peu de définitions relatives l'objet linguistique concerné.

Nous remarquons que les *entités nommées* paraissent se situer à un niveau intermédiaire entre le syntagme (groupe syntaxique) et l'énoncé. Par ailleurs, ces expressions semblent fortement liées à un mécanisme de désignation d'objets ou de concepts. Diverses notions liées aux *entités nommées* étant ainsi évoquées, nous cherchons dans ce chapitre à situer cet objet linguistique au regard de théories linguistiques associées, avant de proposer une définition qui vise à concilier les notions linguistiques recouvertes et les besoins applicatifs nécessitant de distinguer ces expressions particulières du langage naturel.

## 2.2 Les *entités nommées*, un concept émergeant

### 2.2.1 Du signe linguistique à la référence

Les *entités nommées* s'appuient sur les réflexions qui établissent un lien entre le langage comme ensemble de *symboles* signifiants et les objets ou concepts du monde réel que le langage *référence*. De ce point de vue, diverses théories évoquent un lien reposant, selon, sur le *sens*, la *dénotation*, la *référence*, la *désignation*, etc.

Le mathématicien Frege est le premier à établir une distinction claire entre le *sens* et la *référence* [Frege, 1892]. La référence pointe vers un *concept*, qui peut correspondre à un objet du monde réel. De manière plus abstraite, le *sens* est un mécanisme par lequel un *signe* (symbole, nom propre par exemple) peut *désigner* une ou plusieurs référence(s). Il peut y avoir plusieurs sens désignant une même référence, ou a contrario certains sens ne désignant aucune référence. En outre, Frege tient également compte du fait que le sens est nécessairement lié à une *représentation individuelle*, chaque humain interprétant les signes selon son expérience personnelle. Il doit donc exister une convention permettant à plusieurs individus d'attribuer un sens similaire à des expressions complexes du langage naturel. L'extrait suivant résume sa pensée :

*"A proper name (word, sign, sign combination, expression) expresses its sense, stands for or designates its reference. By means of a sign we express its sense and designate its reference."* / *"Un nom propre (mot, signe, combinaison de signes, expression) exprime son sens, tient lieu de ou désigne sa référence. A l'aide d'un signe, nous exprimons son sens et nous désignons sa référence."* [Frege, 1892]

Ainsi Frege propose, par ce qu'il appelle *sens*, un mécanisme autonome régissant l'interprétation du langage naturel, quels que soient les objets supposés ou avérés du monde réel désignés. Ce à quoi le logicien Russel s'oppose [Russell, 1905] en indiquant que, dans certaines situations, il peut ne pas exister de sens à un énoncé, en particulier lorsque l'une de ses composantes ne peut être interprétée. Il introduit à cet effet la *dénotation*, associée aux expressions *non-verbales* (qui ne contiennent pas de verbe) et postule que le sens émerge lors d'une verbalisation incluant ces expressions :

*"denoting phrases never have any meaning in themselves, but [...] every proposition in whose verbal expression they occur has a meaning"* / *"les expressions dénotantes n'ont pas de sens en elles-mêmes, mais [...] chaque proposition dans lesquelles elles sont verbalisées a un sens"* [Russell, 1905]

Ainsi, le sens émerge lors de la prise en compte de l'intégralité d'une proposition. Lorsque nous extrapolons cette théorie, nous remarquons que Russel conditionne l'existence du *sens* à la présence de ce que nous pouvons associer à une *prédication* verbale. Il appuie son propos à l'aide de l'énoncé suivant :

*"the king of France is bald"* / *"Le roi de France est chauve"* [Russell, 1905]

Remarquons qu'à la date où cet exemple est énoncé, en 1905, il n'existe pas de roi en France. Or selon la théorie de Frege, cet énoncé peut avoir un sens alors qu'un de ses composants n'a pas de référence. Pour Russel, la proposition sera interprétée comme fausse, un de ses composants n'ayant pas de dénotation. En outre, par son approche imprégnée de logique, Russel rend centrale la *quantification* implicite du langage naturel dans ce

mécanisme de *dénotation*. Les expressions non-verbales désignent (ou *identifient*) un objet ou un concept par dénotation, et ce mécanisme doit pré-exister à l'élaboration du sens. En outre, diverses expressions peuvent être comparées d'après leurs dénotations. Dans le cas d'expressions complexes, elles sont aujourd'hui couramment appelées *description définies*, comme par exemple :

*"the center of mass of the solar system at the beginning of the twentieth century"* / *"le centre de gravité du système solaire au début du vingtième siècle"* [Russell, 1905]

Nous voyons dès lors l'apparition de deux problématiques distinctes au sein du langage naturel : celle de la désignation pour les expressions non-verbales, celle du sens pour les propositions contenant une prédication verbale. Les vues de Frege et de Russel sont reprises par les travaux de Peirce, puis de Ogden et Richards [Ogden et Richards, 1923], qui proposent une triade : le *signifiant*, le *signifié* et le *référent*. Ces derniers organisent ces trois notions à l'aide du *triangle sémiotique* que nous présentons en figure 2.1, dans lequel le signifié est un intermédiaire qui représente un lien entre le langage (*signifiants*) et des concepts mentaux (*référents*).

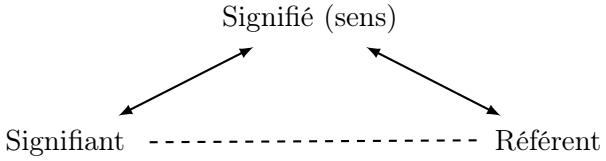



FIGURE 2.1 – Triangle sémiotique

Dans le cadre de nos travaux, nous mettons à part les mécanismes liés à la prédication et nous nous focalisons sur la relation qui s'établit entre le signifié et le référent pour les expressions non-verbales. Intuitivement, il semble que certaines expressions soient quasiment indissociables d'objets mentaux associés, dont les noms propres, lorsqu'ils ne sont pas ambigus (par exemple, *Georges Pompidou*). D'autres ne paraissent pas avoir de référent explicite hors contexte, comme les noms communs introduits par des déterminants indéfinis (par exemple, *un président*). Entre ces deux extrêmes, une multitude d'expressions sont à considérer (par exemple : *'le directeur du département d'informatique de l'Université de Tours'*), qui désignent manifestement un objet mental précis, sans que l'on aie systématiquement besoin d'en déterminer le référent.

Parmi les nombreux travaux sur le sujet, Kripke offre une théorie éclairante [Kripke, 1972]. Par une étude détaillée les mécanismes de désignation, il écarte la possibilité d'identifier un objet à l'aide de ses propriétés ou de son origine historique. Ces éléments, bien qu'objectifs, ne lui paraissent pas suffisants pour constituer le mécanisme par lequel une expression linguistique est spécifiquement liée à un objet mental donné.

Pour Kripke, la relation entre une expression et un référent implique la création d'un *désignateur rigide*, initialisé lors d'un *baptême initial*. Cette théorie met en lumière un mécanisme de désignation qui se veut stable, indépendamment des propriétés connues pour l'objet considéré ou de la nature de l'expression qui est utilisée pour désigner. Il est cependant regrettable que les conditions dudit baptême soient peu explicitées, même

s'il évoque les conventions sociales par lesquelles les hommes s'accordent sur l'objet que doit désigner une expression. Si Kripke n'a pas, à notre connaissance, utilisé le terme d'*entités nommées*, sa théorie y a été associée a posteriori, car elle couvre les noms propres et descriptions définies désignant de manière rigide des objets mentaux, expressions qui paraissent correspondre précisément à celles que certaines tâches TAL cherchent à localiser en priorité.

Pour le cas particulier des noms propres, nous reprenons à notre compte les travaux de Friburger [Friburger, 2002], qui met en avant l'existence d'un *continuum* entre ces derniers et les syntagmes nominaux. Effectivement, la frontière entre ces deux notions est difficile à déterminer lorsque l'on tient compte des phénomènes d'antonomases (transformations d'un nom propre en nom commun et inversement : *un frigidaire*, *un eldorado*, *la Révolution Française*, *le Nouvel An*, etc.), qui se produisent quotidiennement. Ainsi, la seule particularité que nous relevons à leur sujet porte sur leur appartenance à une catégorie *ouverte* et nous supposons en conséquence qu'il est intrinsèquement difficile d'en établir une liste exhaustive.

En somme, ces divers travaux sur le langage naturel permettent de distinguer, au sein d'énoncés, des expressions linguistiques *non-verbales* qui *désignent* de manière rigide des objets mentaux. Comme illustration, considérons par exemple l'énoncé :

*'Georges Pompidou a été président de la république pendant presque cinq années.'*

Outre le nom de personne célèbre *'Georges Pompidou'*, nous y relevons l'entité nommée *'pendant presque cinq années'* qui, malgré sa formulation imprécise, se rapporte manifestement à un intervalle de temps spécifique. S'il est possible de situer précisément cet intervalle (du 20 juin 1969 au 2 avril 1974), il n'apparaît pas nécessaire de faire appel à cette connaissance pour convoquer le même objet mental avec une expression telle que *'durant le mandat de Pompidou'*. Malgré les doutes que nous pouvons émettre (sur les divers mandats de Georges Pompidou, sur les personnes nommées Pompidou, sur les interprétations d'un mandat comme intervalle, etc.), il semble que, lorsque les ambiguïtés sont levées, une désignation *stable* existe entre une expression et un concept et qu'elle a vocation a prendre part au sens, donné par des *opérations* de prédication, à des propositions et à des énoncés.

### 2.2.2 Les entités nommées pour traiter automatiquement le langage

Dans les théories linguistiques présentées jusque là, hormis les travaux récents de Friburger, remarquons que le terme *"entité nommée"* n'a, à notre connaissance, été mentionné par aucun des auteurs. Effectivement, ces travaux étudient des *signifiants*, des *dénotations*, des *désignateurs rigides*, parfois des *entités*, ou plus simplement des *mots*, des *expressions* et des *propositions*. Mais aucun ne fait état d'un objet linguistique qui soit appellé *"entité nommée"*.

De fait, le terme d'*entité nommée* est apparu dans les années 90, à l'occasion de travaux scientifiques tournés vers la *recherche d'information* (RI). A la marge des courants linguistiques de l'époque, cette discipline s'est popularisée suite à la généralisation progressive de l'utilisation d'outils informatiques pour manipuler et échanger des textes (courriers, rapports, presse, documentation, publication scientifique, textes législatifs, loisirs numériques, etc.). Face aux volumes toujours plus importants de données textuelles à traiter, la

recherche automatisée et exhaustive d'information est devenue un enjeu considérable.

Un programme de recherche scientifique portant sur cette thématique a alors été déployé aux États-Unis (*TIPSTER Text Program*, par le *DARPA* [1], 1991-1998) visant à promouvoir le développement de technologies appropriées. Au sein de ces programmes ont été organisées les *campagnes d'évaluations MUC*, financées et encadrées par le gouvernement des États-Unis, réunissant scientifiques et industriels autour de jeux de données et de tâches à résoudre. C'est au fil de ces campagnes, et plus précisément lors de MUC-6 (1993-1996), qu'a émergé le terme d'*entité nommée* (*"named entity"*) [Grishman et Sundheim, 1996]. Aucune définition formelle n'en est donnée, mais le terme est mentionné en lien avec la tâche décrite comme suit :

*"identifying the names of all the people, organizations and geographic locations in a text"* / *"identifier les noms de toutes les personnes, organisations et lieux géographiques dans un texte"* [Grishman et Sundheim, 1996]

Dans ce contexte, trois types d'expressions linguistiques doivent être reconnues : *ENAMEX* (*"entity name expression"*), *TIMEX* (*"time expression"*) et *NUMEX* (*"numerical expression"*). Ainsi, par le terme *entity*, les organisateurs de MUC-6 se focalisaient initialement sur les personnes, organisations et lieux géographiques. Mais la locution *"named entity"* a ensuite été couramment utilisée pour englober les expressions de temps et les expressions numériques, avant de devenir générique pour un ensemble d'objets linguistiques dont on cherche à disposer dans le cadre d'un processus en RI sur des documents textuels écrits.

Par la suite, de nombreuses campagnes d'évaluation poursuivant les mêmes objectifs pour divers langages et types de documents ont eu lieu, dont IREX (*Information Retrieval and Extraction Exercise*, 1999, japonais écrit [2]), CoNLL (*Language-Independent Named Entity Recognition*, 2003, anglais et allemand écrit [3]), ACE (*Automatic Content Extraction*, 2008, anglais écrit [4]), ESTER2 (*Évaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques*, 2009, français oral transcrit [5]), etc.

Nous ne faisons pas ici de description détaillée de ces programmes de recherche, ce travail ayant déjà été réalisé exhaustivement par [Nadeau et Sekine, 2007], mais en donnons un tableau récapitulatif en table 2.4.3. En pratique, ces programmes ont amené à élargir progressivement le champ des expressions comprises comme *"entités nommées"*. Le constat est que nous sommes confrontés ce qui semble être un artefact, créé et profilé au gré des besoins en RI (dont nous illustrons les réalisations les plus courantes en section 2.3). Tout ceci n'a heureusement pas empêché d'établir des liens, au fur et à mesure, parfois a posteriori, avec des théories linguistiques (noms propres, descriptions définies) et sémiotiques (entités, modes de désignation, représentations, ontologies).

Les *entités nommées* ayant été guidées par les besoins des applications informatiques en TAL et en RI, ces objets sont devenus partie des analyses, par exemple au sein de processus incrémentaux tel que nous les avons décrits en section 1.3. De plus, comme

1. Defense Advanced Research Projects Agency
2. `http://nlp.cs.nyu.edu/irex/index-e.html`
3. `http://www.cnts.ua.ac.be/conll2003/ner/`
4. `http://www.itl.nist.gov/iad/mig/tests/ace/`
5. `http://www.afcp-parole.org/camp_eval_systemes_transcription/`

nous l'avons indiqué en section 2.2.1, les expressions linguistiques que nous considérons comme entités nommées ne contiennent a priori pas de prédication verbale. Ainsi, nous postulons alors que la reconnaissance des entités nommées n'a pas nécessité à être réalisée à partir d'une analyse syntaxique. Les traitements qui portent sur les entités nommées nous paraissent devoir se situer entre l'analyse morpho-syntaxique et l'analyse syntaxique. Notons cependant, comme la possibilité en a été évoquée en section 1.3, que les analyses de plus haut niveau, en particulier à l'interface entre la syntaxe et la sémantique (par exemple la catégorisation verbale [Messiant *et al.*, 2010]) peuvent rétroagir sur des analyses préalables (dont la syntaxe peut contraindre des analyses en entités nommées ambiguës).

Nous nous appuyons sur les travaux récents d'Ehrmann [Ehrmann, 2008], qui fait état de la place qu'occupent les entités nommées au sein d'un processus TAL motivée par des considérations linguistiques. La rétrospective des théories linguistiques en jeu est particulièrement approfondie et détaillée et y décrit les expressions linguistiques concernées au sein des syntagmes nominaux : noms propres et descriptions définies. De plus, Ehrmann nous propose la définition suivante :

*"Étant donné un modèle applicatif et un corpus, on appelle entité nommée toute expression linguistique qui réfère à une entité unique du modèle de manière autonome dans le corpus."* [Ehrmann, 2008]

De notre point de vue, une *"expression linguistique qui réfère à une entité unique du modèle"* peut-être mise en correspondance avec la notion de *désignateur rigide* introduite par Kripke. Nous reprenons à notre compte cette partie de la définition en l'interprétant comme une *désignation stable*. Cependant, pour Ehrmann, la définition des entités nommées est conditionnée à l'existence d'un *"modèle applicatif"* et d'un *"corpus"*. Un processus TAL serait indispensable pour que les entités nommées puissent s'y inscrire. Nous mettons en cause ce présupposé et, s'il semble que des visées *applicatives* ont fait émerger le concept d'entité nommée, nous ne voyons pas de nécessité à disposer d'un modèle ou d'un corpus pour les définir de manière générale. Notre hypothèse sera plutôt que ces expressions ont vocation à être utilisées par des applications et sont manipulées par ces dernières afin d'élaborer des représentations, par exemple sous forme de prédications dont elles seraient exclusivement des arguments. De plus, comme nous le verrons plus loin, nous nous interrogeons également l'*autonomie* de ces expressions linguistiques.

Si les entités nommées sont sollicitées pour des visées applicatives, il semble cependant difficile de délimiter le champ des applications concernées. Ceci a été illustré lors d'une journée d'étude dédiée aux entités nommées [Nouvel *et al.*, 2011b] [6], où l'on a vu que ces objets sont utiles à des applications très diverses. De même, au sein du projet *Quaero* [Galibert *et al.*, 2011], la tâche concernant les entités nommées, se voulant généraliste, les définit alors de la manière suivante :

*"mono- or multi-word expression belonging to a potentially interesting class for an application"* / *"expression mono- ou multi-mots appartenant à une classe potentiellement intéressante pour une application"* [Galibert *et al.*, 2011]

Au sein du guide d'annotation Quaero [7], nous trouvons les définitions suivantes :

*"les entités nommées incluent traditionnellement trois grandes classes : les noms, les*

6. Organisée par l'ATALA, Association pour le Traitement Automatique de LAngues
7. `http://quaero.org/media/files/bibliographie/quaero-guide-annotation-2011.pdf`

*quantités, les dates et durées. Nous nous plaçons dans le contexte d'extraction d'information (entités, relations) servant à constituer une base de connaissances."*

Il semble à nouveau important que les entités nommées soient représentées comme objets que diverses applications auront à leur disposition. A la manière dont Russel distingue les expressions non-verbales et les propositions logiques, nous émettons l'hypothèse que deux procédés peuvent être distingués : le premier détermine des objets mentaux à partir d'expressions linguistiques (dont les entités nommées), le second liant ces objets au sein de formules logiques (dont les propositions et les énoncés). La perspective dans laquelle nous nous situons construit ces représentations dynamiquement, et de ce point de vue nous nous inspirons des *"grammaires instructionnelles"* [Col *et al.*, 2010], qui distinguent *entités* et *procès* à l'aide de la définition suivante :

*"une entité est topologiquement individuable et stable dans son évolution sur la scène, alors qu'un procès présente les relations entre les entités et ces relations évoluent dans la temporalité de la scène"* [Col *et al.*, 2010]

Dans notre travail, nous ne nous préoccupons pas directement de la construction du *procès* (ou *scène verbale*, assimilée aux propositions et énoncés), mais retenons le caractère *stable* des entités lors de la construction d'une représentation. Par ailleurs, nous reprenons également à notre compte le principe d'une mécanique guidée par des *instructions dynamiques* sous-jacentes au langage naturel. Nous considérons que les instructions liées à la constitution des entités sont élémentaires, à la différence des opérateurs logiques (par exemple la prédication verbale) qui construisent ultérieurement une représentation du sens d'un énoncé donné.

En conclusion, nous avons fait état dans ce chapitre d'un ensemble de théories qui nous paraissent fortement liées à la compréhension de la réalité linguistique des entités nommées comme expressions particulières du langage naturel. Jusque là, notre positionnement reste cependant abstrait : il doit se fonder sur la réalité des problématiques liées aux entités nommées. Après ces considérations théoriques liées à la manière dont le concept d'entités nommées à émergé, décrivons maintenant plus en détail à quoi les entités nommées sont utiles et quels phénomènes linguistiques peuvent les rendre difficiles à reconnaître.

### 2.2.3 Quelques problématiques liées aux entités nommées

Comme nous l'avons vu, la notion d'entité nommée est mouvante et fait appel à de nombreux domaines : théorie de la désignation, noms propres, descriptions définies, modèles applicatifs en RI, analyse incrémentale du langage, etc. D'une part, détecter, reconnaître et résoudre les entités nommées, même si leur champ était clairement défini, présente encore des difficultés majeures dans certains cas (entités peu courantes, entités nouvelles dans l'actualité, variantes d'écriture lors de leur traduction, etc.). D'autre part, ces difficultés sont à chaque fois plus saillantes alors que le périmètre recouvert est continuellement étendu (adresses, produits, événements, etc.). Le besoin de clarifier cette notion et de disposer d'un module de traitement dédié à leur sujet se fait alors de plus en plus ressentir, par exemple pour les applications suivantes :

– **Indexation et recherche d'information** : les entités nommées *détectées* dans des documents peuvent permettre de construire des index que pourront exploiter les

moteurs de recherche.
- **Annotation en rôles sémantiques** : dans le cadre d'un mécanisme de compréhension, déterminer les rôles (agent, patient, objet, instrument, lieu, destination, etc.) peut être conditionné par les types d'entités nommées *reconnues*.
- **Question - réponse** : le mécanisme par lequel une machine fournit une réponse à une question donnée peut nécessiter de *résoudre* des entités dans la question, afin de rechercher la réponse dans des bases de connaissances.
- **Résolution conjointe d'autres tâches TAL** : tokenisation, analyse morpho-syntaxique ou syntaxique, reconnaissance de l'écriture et de la parole, résolution d'anaphores [Lin *et al.*, 2010] sont des tâches qui peuvent interagir avec la détection ou la reconnaissance des entités nommées.

Ces applications, idéalement, nécessiteraient que toutes les entités nommées soient au préalable automatiquement résolues (donc détectées et reconnues). Les nombreux travaux sur le sujet montrent que les systèmes dont nous disposons ont des difficultés à le faire dans le cas général. Considérons les problématiques de reconnaissance et de résolution des entités suivantes, volontairement difficiles :

- (1) *'Au Tibet,* ***Tenzin Gyatso*** *est très respecté [...]'*
- (2) *'L'élection de* ***Georges Bush*** *en 2004 [...]'*
- (3) *'Étant donné que* ***Berlin*** *n'a pas ratifié [...]'*
- (4) *'La contribution de* ***Hautecloque*** *pour vaincre l'****Allemagne*** *est reconnue [...]'*

L'exemple (1) contient une entité qui peut difficilement être résolue sans connaissances exhaustives, *'Tenzin Gyatso'* étant le nom du *14 ème Dalaï Lama*. Il reste néanmoins possible de reconnaître cette entité (en tant que personne) à l'aide du contexte *'est très respecté'*. L'exemple (2) illustre une ambiguïté lors de la résolution de l'expression *'Georges Bush'* (père ou fils), qui peut être levée grâce à des connaissance extérieures sur les candidats d'élections en 2004. Cependant, la *reconnaissance* de cette entité comme une personne paraît peu problématique, à l'aide du contexte *'élection de'* et des composants du nom propre *'Georges'* et *'Bush'*. Pour (3), une interprétation est nécessaire pour faire référer *'Berlin'* au gouvernement allemand. Mais lorsque la reconnaissance de l'entité comme organisation (administration) est réalisée, sa résolution au gouvernement allemand fait appel à peu de connaissances. Enfin, (4) nous présente un condensé de ces difficultés, dans lequel *'Hautecloque'* doit être résolu au Général Leclerc, tandis que l'*'Allemagne'* correspond l'armée du Troisième Reich.

Au travers de ces exemples, nous illustrons les difficultés occasionnées par les phénomènes linguistiques suivants :

- **Synonymie** : plusieurs expressions peuvent désigner un même référent (*'Ville-Lumière'*, *'Paris'*, *'capitale française'*, *'ville de la Tour Eiffel'*, etc.)
- **Homonymie** : des référents distincts sont désignés par une même expression (*'Charles de Gaulle'* pour le personnage, un aéroport, un porte-avion, des avenues, etc.)
- **Métonymie** : en contexte, une expression est utilisée pour désigner un référent qui lui est proche (la *'France'* pour l'équipe de football, un *'Picasso'* pour une de ses œuvres, *'Washington'* pour le gouvernement des États-Unis, etc.)

En élargissant la problématique de la synonymie, nous constatons que, pour un référent donné, toutes les expressions le désignant ne sont pas nécessairement connues a priori.

Nous sommes amenés à supposer l'existence d'un mécanisme qui permet, dans certaines situations, de reconnaître ou même de résoudre des entités nommées à l'aide d'indice partiels de l'entité ou dans son contexte. Pour les cas d'homonymie, une ambiguïté est a priori connue, mais, comme l'expression linguistique est identique pour plusieurs référents, seul le contexte pourra aider à la lever. Enfin, pour la métonymie, la désignation réalisée par une expression linguistique est de toute évidence contrainte par le contexte [Markert et Hahn, 2002, Poibeau, 2006].

Dans le cadre des théories linguistiques que nous avons exposées en section 2.2.1, nous voyons resurgir la problématique de la désignation et de la relation entre signifiants et référents. Nous constatons cependant que la désignation par une expression linguistique paraît n'être qu'en partie liée au signifiant lui-même. Dans de nombreux cas, le contexte influence fortement la reconnaissance et la résolution des entités nommées. En ceci, nous nous voyons contraints de réfuter l'*autonomie* de l'entité en tant qu'objet linguistique, comme Ehrmann le propose. Les connaissances a priori sur les expressions linguistiques considérées comme entités nommées sont indispensables, mais paraissent bien insuffisantes pour les reconnaître et les résoudre correctement.

Les travaux que nous présentons ici portent sur la reconnaissance des entités nommées en cherchant à déterminer leurs frontières et leur type. Nous avons à disposition, comme indiqué en section 1.3, des lexiques et des transducteurs qui contiennent de nombreuses expressions liées aux entités nommées (dont des listes assez conséquentes de noms propres). Nous prenons le parti de supposer que ces ressources sont suffisamment exhaustives pour nos besoins immédiats (même si nous les enrichissons régulièrement par ailleurs). Ainsi, nous nous focalisons plutôt sur la collecte automatique, dans des corpus volumineux, d'indices dont nous observons qu'ils participent à la reconnaissance des entités nommées, sans distinguer spécifiquement les cas les plus simples des cas de synonymies, d'homonymies ou de métonymies.

## 2.3 Les typologies d'Entités Nommées

### 2.3.1 Formes de typologies

Réaliser la reconnaissance d'entités nommées suppose de, simultanément, les détecter et leur associer un type. Pour ce faire, il faut au préalable disposer d'une typologie au sein de laquelle les types appropriés pourront être sélectionnés. Les typologies construites à cet effet peuvent avoir une plus ou moins grande complexité. Dans la plupart des travaux, les typologies forment une partition des entités nommées selon des critères d'ordre sémantique. Les types peuvent ensuite être organisés, sous forme de *hiérarchies* ou d'*ontologies*. Ceci peut être formalisé comme une relation d'ordre entre ces types. Dans la plupart des cas, les entités nommées sont généralement associées aux *feuilles* de cette typologie, plus rarement à n'importe quel nœud.

Nous ne discutons pas en détail ces considérations, mais notons néanmoins qu'il pourrait s'avérer nécessaire à l'avenir, dans des situations d'ambiguïté ou de rôles multiples du référent, de tenir compte d'une plus grande finesse pour la reconnaissance des entités nommées, par exemple à l'aide de :

- **Types généraux** : une entité nommée peut être associé à un *nœud non feuille* de la typologie.
- **Types multiples** : une entité nommée peut être associée à plusieurs types (*facettes*) au sein de la typologie.
- **Treillis** : la typologie est organisée comme un *treillis* plutôt qu'une hiérarchie : un type peut avoir plusieurs sur-types.
- **Structures de traits** : à une entité nommée peuvent-être associés un ensemble de traits, un raisonnement ultérieur permettant de déduire, pour une entité, son ou ses type(s) selon ses traits définis.

Afin de rester cohérents avec les expérimentations que nous réalisons, nous reconnaissons les entités nommées selon une typologie similaire à celle décrite au sein du projet *Quaero* [Galibert *et al.*, 2011]. Les entités nommées y sont réparties en 7 types primaires et 32 sous-types, dont voici la liste exhaustive :

- **Personnes** : personnes individuelles, personnes collectives.
- **Lieux** : lieux administratifs (ville, région, nations, supranations), lieux physiques (géographiques, hydrologiques, astrologiques).
- **Organisations** : entreprises, administrations.
- **Temps** : dates (absolues ou relatives) et horaires (absolus ou relatifs).
- **Montants** : quantités, durées.
- **Produits** : objets manufacturés, œuvres artistiques, œuvres médiatiques, produits financiers, logiciels, récompenses, voies, doctrines, lois.
- **Fonctions** : fonctions individuelles, fonctions collectives.

Cette typologie ajoute donc aux entités nommées traditionnelles les produits et les fonctions. Elle ajoute une granularité supplémentaire sous-types) aux types principaux (ou *primaires*). Nous notons que les produits recouvrent une assez grande diversité de sous-types. Nous reviendrons sur cette typologie, au regard des données annotées, lors de la présentation du corpus Etape en section 7.3.3.

### 2.3.2 Noms propres de personnes, lieux et organisations

Comme nous l'indiquons en section 2.2.2, les besoins en recherche d'information se sont initialement focalisés sur le traitement des *noms propres*. Ces éléments sont relativement courants dans le langage : dans l'analyse d'une édition du journal *Le Monde*, [Friburger, 2002] fait état de 90 056 formes simples, dont 3 755 (4,16%) noms propres. Cette proportion peut varier et atteindre des proportions plus importantes [Stephens, 1993]. Nous supposons que les corpus issus de transcriptions d'émissions radiodiffusées ou télévisuelles en comportent un nombre suffisamment conséquent pour avoir à les prendre en compte.

Dans le cadre d'un processus TAL, ces éléments demandent à être reconnus le plus tôt possible, afin qu'il soit possible d'y appliquer des traitements particuliers. En premier lieu, la tokenisation doit être en mesure de tenir compte de noms propres qui peuvent être composés de plusieurs tokens, comme les noms de personnes composés de prénoms et de noms. La lemmatisation n'a que peu de sens pour ces unités, ou peut être ramenée à une normalisation si une entité nommée est résolue au sein d'une base de connaissances. Les traitements morpho-syntaxiques nécessitent qu'à ces éléments soit affectée une catégorie

morpho-syntaxique a priori. Enfin, les traitements sémantiques peuvent évidemment tirer partie des nombreuses informations fournies lors de la reconnaissance de noms propres.

Les premiers travaux sur les noms propres se sont concentrés sur la reconnaissance de trois types principaux :

- **personnes** ou *anthroponymes*,
- **lieux** ou *toponymes*,
- **organisations** ou *anthroponymes collectifs*.

Ces trois types d'entités nommées ont en commun de désigner des *objets du monde réel* liés aux activités humaines et couramment référés au sein de textes. Il est possible de les recenser sous forme d'une base de connaissance, comme cela est réalisé dans le cadre du projet Prolexbase [Bouchou et Maurel, 2008]. Cette base lexicale couvrante (et de surcroît multilingue) établit une liste de noms propres et des traits qui peuvent leur être associés. Elle est centrée autour de *prolexèmes* (formes de surface, qui peuvent être fléchies), entre lesquels des relations sont établies (*pivots* d'une langue à une autre, *synonymie*, *méronymie* et *accessibilité*).

Remarquons que, bien que *simples* de prime abord, ces types d'entités nommées sont facilement sujets aux phénomènes d'homonymie, de synonymie et de métonymie. Pour un humain, ces phénomènes sont instantanément reconnus, comme le montre l'exemple de l'entité *'Washington'* au sein des énoncés suivants :

- *'Nous avons été en vacances à* ***Washington*** *et à Boston.'* (*Lieu*)
- *'Hier,* ***Washington*** *a battu New York 34 à 10.'* (*Organisation sportive*)
- *'Lors des discussions,* ***Washington*** *a opposé son veto.'* (*Org. administrative*)
- *'Parmi les pères fondateurs,* ***Washington*** *est le plus connu.'* (*Personne*)

Dans le cadre d'une reconnaissance automatique des entités nommées, ces phénomènes sont loin d'être triviaux pour une machine. Nous notons que la tâche de *détection* n'est que peu impactée, tandis que la *reconnaissance* et la *résolution* demandent un examen systématique du contexte. Nous émettons l'hypothèse que, dans une majorité de cas, le contexte immédiat est suffisant pour détecter ces phénomènes et qu'il n'est pas nécessaire de faire appel à des inférences logiques complexes pour les prendre en compte.

Au sujet des noms propres, précisons à nouveau (c.f. 2.2.1) que nous ne tenons que peu compte de la distinction entre ces derniers et les autres expressions linguistiques. Étant une catégorie ouverte, nous éviterons cependant de prendre en compte des mécanismes spécifiques à un nom propre donné. Les bases lexicales ont nous disposons nous paraissant suffisamment couvrantes, nous substituons alors aux noms propres les traits sémantiques qui y sont associés dans ces bases. Ceci doit nous permettre d'observer des régularités sur la base de ces traits sémantiques, plutôt que des particularités relatives aux noms propres apparaissant dans les corpus que nous utilisons. Lorsque c'est nécessaire, nous pouvons enrichir la base lexicale, en faisant l'hypothèse que les régularités découvertes s'appliqueront à n'importe quelle nouvelle entrée.

### 2.3.3 Expressions de temps, adresses et montants

Les représentations construites à partir d'énoncés peuvent tirer parti d'autres éléments qui désignent de manière plus complexe certains objets mentaux à manipuler. Dans ce

contexte, après les noms propres, les expressions de temps ont été étudiées plus en détail, avec l'objectif d'extraire des informations à partir de textes. Leur forme peut-être très diverse, comme par exemple :

- *'Noël'*, *'Pâques'*, *'le jour de l'An'*
- *'le 9 septembre 2001'*, *'le 21 avril 2002'*, *'le 1 $^{er}$ mai'*, *'le 31 octobre'*
- *'à 15h'*, *'entre 16 et 20h'*, *'en matinée'*, *'ce midi'*
- *'hier'*, *'demain'*, *'mercredi prochain'*, *'l'année dernière'*, *'l'année dernière'*

Nous constatons que ces descriptions définies peuvent être utilisées afin de construire une représentation temporelle associée à un énoncé. Il semble qu'un procédé similaire soit à l'œuvre pour des représentations spatiales dans les expressions suivantes :

- *'place de l'Église'*, *'avenue de la République'*
- *'9 rue Pasteur'*, *'148 rue Saint-Honoré'*
- *'la région parisienne'*, *'l'agglomération lilloise'*
- *'ici'*, *'là-bas'*

Au premier abord, il apparaît que ces modes de désignation nécessitent une étape supplémentaire pour leur compréhension. Effectivement, au lieu d'associer directement une expression linguistique à un objet mental, une concaténation d'éléments au sein des expressions détermine une représentation, par composition. Il reste à déterminer si, dans une optique de reconnaissance, un raisonnement est nécessaire pour déterminer qu'il y a désignation ou pour élaborer une représentation.

De la même manière, l'acception des entités nommées est étendue à divers objets qui se construisent selon un même procédé. Un rapprochement semble pouvoir se faire, dans leurs modes de désignation, entre une *date* (jour, mois, année), une *adresse* (numéro, voie, code postal) et d'autres expressions linguistiques, comme par exemple une mesure physique (poids, volume, etc.) ou une somme monétaire (selon la devise et ses éventuelles subdivisions). Nous considérons alors également les *montants* et *quantités*, comme par exemple :

- *'250 kilomètres'*
- *'2 000 hectolitres'*
- *'15 euros et 30 centimes'*

Nous remarquons alors qu'il existe des mécanismes qui paraissent désigner sans pour autant faire appel à des inférences logiques. Ils semblent être construits par composition d'éléments contigus au sein d'expressions, les objets mentaux correspondants pouvant être reconnus (typés) sans faire appel à un raisonnement complexe. Par ailleurs, dans de nombreux cas, ces expressions ne peuvent former une énoncé à elles-seules : elles paraissent *devoir* être intégrées à une proposition ou à un dialogue, comme par exemple :

*'**la région parisienne** est chère, **ce midi** mon repas m'a coûté **15 euros**.'*

L'énoncé contient deux propositions logiques, au sein desquelles sont mis en relation un lieu, une expression de temps et un montant monétaire. Les entités nommées y désignent des objets mentaux sur lesquels opèrent les propositions. Nous émettons l'hypothèse que les désignations (contenues dans les propositions) ne relèvent pas de prédications ou de quantifications logiques, mais sont construites et mises à disposition par des procédés préalables. En ceci, nous supposons que les objets mentaux associés aux entités nommées existent avant tout raisonnement et sont stables au travers des inférences qui les convoquent.

### 2.3.4 Produits, marques, fonctions, etc.

Enfin, selon les domaines d'applications considérés, les entités nommées peuvent encore recouvrir diverses expressions linguistiques. Il est évident que les noms propres de personnes, lieux ou organisations ne couvrent pas tous les noms propres. Et même s'ils sont très majoritaires pour certains types de textes (journalistiques), on imagine aisément des situations dans lesquelles la reconnaissance aura intérêt à être étendue à d'autres types (*produits*, *marques*, *événements*, *véhicules*, etc.).

En ce qui concerne les descriptions définies, le même constat peut être fait. Si certains types apparaissent plus courants que d'autres, cela n'exclut pas le besoin de prendre en compte toutes les expressions linguistiques qui pourraient, de la même manière, désigner des objets mentaux donnés et stables sur lesquels des raisonnements logiques opèrent. Tel peut effectivement être le cas pour les *fonctions* (humaines), certaines désignation de *bâtiments* ou de *mouvements politiques*, et bien d'autres.

Pour une étude plus exhaustive des diverses catégories que l'on peut chercher à reconnaître en tant qu'entités nommées, nous renvoyons vers [Nadeau et Sekine, 2007]. Cependant, étendre les expressions concernées par le concept d'entités nommées nous paraît peu satisfaisant pour définir ces dernières. En effet, il nous semble raisonnable d'espérer que l'observation systématique des notions qu'elles recouvrent ou des expressions linguistiques qu'elles peuvent revêtir ne soit qu'une étape intermédiaire vers la caractérisation de ces objets linguistiques, quelque soit la diversité des types qu'ils recouvrent.

### 2.3.5 Récursivité et compositionnalité des entités nommées

Certains exemples mentionnés, à des degrés divers, semblent mettre en jeu des entités sous-jacentes, dont il est souvent difficile de considérer la portée. Considérons les exemples suivants :

- *'directeur de la SNCF'*, *'cabinet du préfet de la région Île-de-France'*
- *'Banque de France'*, *'Porte de Clignancourt'*, *'Blanche de Castille'*
- *'Fort-de-France'*, *'Louis-le-Grand'*, *'Amfreville-sur-Iton'*

Nous constatons que pour chaque exemple, des entités nommées peuvent être décelées au sein d'autre entités nommées. Il y a donc dans ces expressions une récursivité dont la portée est difficile à saisir. Certains cas correspondent à des description définies, comme *'cabinet du préfet de la région Île-de-France'*, une organisation, qui contient *'préfet de la région Île-de-France'*, une personne, contenant elle-même *'région Île-de-France'*, un lieu. D'autres sont des locutions, des expressions figées, comme *'Fort-de-France'*, dont les entités sous-jacentes paraissent moins saillantes.

Nous remarquons par ailleurs, comme cela est réalisé au sein du projet *Quaero* [Grouin *et al.*, 2011, Rosset *et al.*, 2012], que l'on peut de manière générale qualifier les éléments qui composent les entités nommées. Ainsi, les prénoms participent à la construction d'entités nommées de personnes, les chiffres à celles des dates ou des montants, etc. Ajoutons que si certaines entités nommées résultent ainsi d'une composition, nous soupçonnons alors qu'il soit possible de *décomposer* leur reconnaissance.

La récursivité et la décomposition ne peuvent se substituer au besoin de disposer des entités nommées les plus larges. Par ailleurs, ces mécanismes paraissent apporter bien peu pour des expressions figées, en particulier lorsqu'elles sont considérées comme un seul token. Mais il peut s'avérer utile, dans certaines situations, de reconnaître les entités ou composants imbriqués, en particulier lorsque cela est nécessaire pour la reconnaissance de l'objet mental considéré. Tel est par exemple le cas pour *'directeur de la SNCF'*, où le composant *'directeur'* et l'entité nommée *'SNCF'* paraissent indispensables pour que la désignation se fasse correctement.

Nous considérons donc pertinent de décrire les récursivités et les compositions d'entités nommées. Une discussion en profondeur sur le sujet nécessiterait cependant d'autres considérations, notamment en morphologie et en traitement des expressions composées, qui sortent du cadre de notre travail. Nous suivons les directives données dans le cadre du projet *Quaero*, qui impliquent la reconnaissance des entités nommées récursives et celles des composants suivants, selon les entités concernés :

- **Toutes entités** : nom, genre, qualificateur, valeur, unité, objet, intervalle.
- **Personnes individuelles** : prénom, nom de famille, titre.
- **Adresses** : numéro, code postal.
- **Expressions de temps** : semaine, jour, mois, année, siècle, millénium, modificateur.

## 2.4 Annotation et évaluation des entités nommées

### 2.4.1 Annotation manuelle de corpus

Pour de nombreuses problématiques linguistiques, relever et analyser les phénomènes à caractériser est réalisé à l'aide d'outils informatiques. Dans un grand nombre de cas, il s'agit de relever les occurrences au sein d'un ensemble de textes, le *corpus*. Ce travail est réalisé lors d'un processus *d'annotation*, qui vise à distinguer et qualifier au sein du corpus des extraits (tokens, expressions, propositions) objets du phénomène à observer. L'annotation de corpus est une thématique très active qui fait l'objet de nombreux travaux. Effectivement, celle-ci peut être plus ou moins assistée, guidée, automatisée. De plus, comme le montre [Brun et Hagège, 2008, Fort *et al.*, 2009], ce travail nécessite une grande rigueur et beaucoup de préparation afin d'obtenir une annotation fiable. Dans l'essentiel, trois éléments paraissent indispensables :

- **Guide d'annotation** : détaille les expressions linguistiques à annoter, selon des critères qui doivent laisser aussi peu de latitude que possible à la personne qui réalisera l'annotation.
- **Outils d'annotation** : logiciels servant à annoter, dont les interfaces doivent faciliter, mais sans biaiser, le travail de l'annotateur, en incluant éventuellement une phase de pré-annotation automatique.
- **Mesures d'évaluation de la qualité des annotations** : tests prévus afin de confirmer la fiabilité (ou d'exhiber l'arbitraire) d'une annotation (accord inter-annotateurs) sur les parties annotées par plusieurs personnes (annotation croisée).

Nous voyons que ce travail doit déjà satisfaire une exigence d'automatisation. Effectivement, un processus d'annotation non-arbitraire doit, en théorie au moins, être tota-

lement déterminé par les informations contenues dans le guide. L'annotation relève alors d'un mécanisme *automatique* dans lequel n'importe quel être humain disposant du guide d'annotation (et de connaissances générales suffisantes selon le phénomène à annoter) est supposé prendre la même décision dans les mêmes circonstances. Dès lors, nous voyons que le nombre de conditions à remplir pour que la qualité d'une annotation soit jugée satisfaisante est difficile à atteindre. Ceci peut conduire à des annotations d'une fiabilité moindre, comme nous le verrons en 7.3 à propos des jeux de données que nous utilisons.

Le guide précise les possibilités et les limites pour annoter les entités de manière générale, pour n'importe quel texte. En premier lieu y sont indiqués les types d'entités à annoter (dont nous donnons des exemples en section 2.3). Également, lorsque des portions de textes peuvent recevoir plusieurs annotations, les structures possibles (imbrications, chevauchements, types multiples) sont généralement contraintes et des directives sont données pour résoudre les cas problématiques. Enfin, si nécessaire, le guide d'annotation peut prévoir la mise en place d'un *référentiel* (base de données, de connaissances). Ceci est plus particulièrement utile lorsqu'il s'agit de *résoudre* les entités nommées, ce qui nécessite de faire pointer les entités vers des objets du monde réel *référencés*. Notons que le processus d'annotation est défini indépendamment de l'annotateur (humain ou machine). La figure 2.2 donne un aperçu des éléments en jeu dans ce processus.

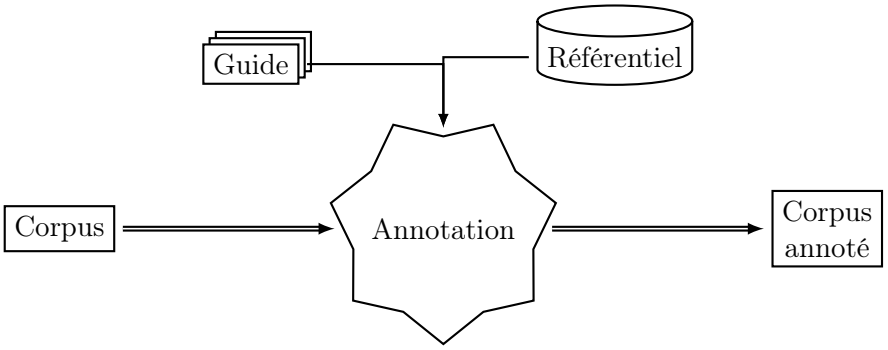



FIGURE 2.2 – Éléments d'un processus d'annotation

En ce qui concerne l'annotation des textes, parmi les codes couramment utilisés pour relever des occurrences dans un corpus, nous mentionnons les suivantes pour l'exemple *'Le président Georges Pompidou'* (où *'Georges Pompidou'* est une personne, abrévié `'PERS'`) :

- **Étiquetage** : `'Le président {Georges Pompidou,+PERS}'`
- **Parenthésage** : `'Le président [PERS Georges Pompidou]'`
- **Balisage** : `'Le président <PERS> Georges Pompidou </PERS>'`
- **Classification** : `'Le président Georges/PERS Pompidou/PERS'`

Nous n'entrerons pas ici dans les détails technique pour enregistrer ces divers codes sur support informatique. Notons simplement que, dans la plupart des cas, modulo quelques adaptations si nécessaires, ces formats peuvent être rendus compatibles les uns avec les autres. Par ailleurs, les annotations peuvent être mélangées au texte (intégrées) ou reportées dans des fichiers à part qui indiquent les positions de chaque annotation au sein du texte (déportées).

Dans ce manuscrit, les annotations sont indiquées par un balisage *SGML* par utilisation de chevrons `'<'` et `'>'` et de la barre oblique `'/'` pour différencier la balise fermante. Par exemple, `'<personne>'` marque le début d'une entité de type personne, `'</personne>'` en marque la fin. Annoter le nom propre *'Pablo Picasso'* comme personne est noté `'<personne> Pablo Picasso </personne>'`. Pour plus d'uniformité, les catégories d'entités sont écrites en anglais et abréviées : `'pers'` pour les personnes, `'org'` pour les organisations, `'loc'` pour les lieux, `'time'` pour les expressions de temps, `'amount'` pour les montants, etc. Selon le contexte, l'imbrication d'annotation pourra être possible (par exemple : `'Le <org> Centre <pers> Pompidou </pers> </org>'`), mais, dans le cas général, ne pourront être discontinues ni se chevaucher.

### 2.4.2 Métriques d'évaluation

La reconnaissance des entités nommées, comme nous l'avons indiqué en section 2.2.2, est une problématique qui émergé de travaux en recherche d'information. Selon le paradigme dans ce domaine, les mesures habituellement utilisées pour évaluer la performance d'une recherche au sein d'une base de documents ont été initialement adaptées aux entités nommées. Celles-ci mesurent le nombre d'objets pertinents effectivement récupérés parmi un ensemble d'objets, et en déduisent la précision, le rappel et la f-mesure [van Rijsbergen, 1979] de l'application considérée. Nous formulons ici le calcul de ces mesures pour la reconnaissance d'entités nommées.

Soit $D$ un document, pour lequel a été appliquée une tokenisation, ou segmentation en unités minimales. Nous considérons alors ce document comme une séquence de $n$ items (correspondants aux *tokens*) $D = \langle d_1, d_2 \dots d_n \rangle$. Le processus d'annotation peut alors être considéré comme l'affectation d'une catégorie à chaque token parmi les $k$ catégories d'entités nommées possibles $c_1 \dots c_k$. De surcroît, nous notons $\emptyset$ la catégorie *vide*, i.e. le fait de n'avoir aucune annotation pour un token donné.

Dans ce contexte, nous confrontons deux annotations : celle réalisée par un annotateur humain qui établit une *référence* $R$, et une annotation qualifiée d'*hypothèse* $H$, dont nous cherchons à évaluer la qualité. La première sera notée $\langle r_1 \dots r_n \rangle$ et la seconde $\langle h_1 \dots h_n \rangle$. Ainsi, $c_j \in r_i$ indique que, dans l'annotation de référence, la catégorie $j$ a été attribuée au token $i$. De plus, $|\{i, c_j \in h_i\}|$ est le nombre de tokens (de l'hypothèse en l'occurrence) pour lesquels la catégorie $j$ a été affectée.

Les métriques habituelles en extraction d'information peuvent alors être écrites sous la forme suivante :

– **Précision** :
$$Pre(R,H) = \frac{|\{i | r_i = h_i, r_i \neq \emptyset\}|}{|\{i | h_i \neq \emptyset\}|}$$

– **Rappel (en anglais, *"recall"*)** :
$$Rec(R,H) = \frac{|\{i | r_i = h_i, r_i \neq \emptyset\}|}{|\{i | r_i \neq \emptyset\}|}$$

– **F1-mesure** :
$$Fm(R,H) = \frac{2 * Pre(R,H) * Rec(R,H)}{Pre(R,H) + Rec(R,H)}$$

Lors de ce calcul, pour être jugés corrects, les tokens doivent recevoir en hypothèse *toutes* et *seulement* les annotations qui ont été attribuées en référence. Si ces métriques ont certes l'avantage de faire un lien direct avec l'extraction d'information (par exemple recherche de documents correspondant à une requête), elles ont le désavantage d'être bien peu tolérantes aux erreurs. En particulier, lorsque des entités sont imbriquées, l'absence de l'entité la plus large invalide toute l'hypothèse, même si des entités internes ont été correctement reconnues.

D'autre métriques ont alors été proposées, la plupart calculant une *"pondération d'erreurs"* de l'hypothèse au vu de la référence [Makhoul *et al.*, 1999]. A cet effet, la tokenisation, supposée similaire entre l'hypothèse et la référence, permet de mettre efficacement ces deux annotations en correspondance. Ce calcul est donc réalisé à partir des entités (ou *"slots"* en anglais), en essayant de déterminer si les entités en hypothèse peuvent être associées à des entités en référence, partiellement (entités générant des erreurs de type ou de frontière) ou totalement (entités correctes).

Cette mesure, dont le calcul est maintenant maîtrisé (voir par exemple [Galibert *et al.*, 2011]), permet alors de considérer les types d'erreurs suivants :

- **Déletion** ($Err_D$) : aucune entité en hypothèse, une entité en référence.
- **Insertion** ($Err_I$) : une entité en hypothèse, aucune entité en référence.
- **Type** ($Err_T$) : deux entités de types différents en référence et en hypothèse.
- **Extension** ($Err_E$) : deux entités de même types en référence et en hypothèse, mais elles ne commencent ou ne se terminent pas aux mêmes positions.
- **Type et extension** ($Err_{TE}$) : combinaison des deux erreurs précédentes.

Il devient alors possible d'évaluer la qualité d'une annotation en choisissant, parmi les appariements possibles, celui qui minimise cette erreur. De plus, de part et d'autre, une entité ne peut être appariée qu'avec au maximum une entité. Ainsi, une entité pourra être parfaitement appariée (pas d'erreur), partiellement appariée (erreur $T$ ou $E$), ou pas appariée du tout (erreur $D$ ou $I$). Finalement, ces erreurs peuvent être pondérées et ramenées au nombre $R$ d'entités présentes dans la référence, nous obtenons le Slot Error Rate (taux d'erreur par *élément*) que, dans notre travail, nous utilisons avec la pondération suivante :

$$SER(R,H)) = \frac{|Err_D| + |Err_I| + |Err_{TE}| + 0.5 * |Err_T| + 0.5 * |Err_E|}{|Slots(R)|}$$

Nous remarquons que ce taux d'erreur pénalise moins les erreurs de type ou d'extension. Intuitivement, ceci correspond au fait que l'on récompense le fait de ne reconnaître que partiellement (type ou frontière) une entité nommée. De plus, cette mesure ne tient pas compte de la longueur (en tokens) des entités, par exemple ne pas reconnaître une date qui concerne fréquemment plusieurs mots (*'le 3 janvier 2012'*) ne sera pas plus coûteux que de ne pas reconnaître un lieu (*'Paris'*). Nous obtenons alors une mesure qui met la priorité sur la détection des entités nommées, en étant moins pénalisante pour une erreur de reconnaissance sur une entités détectée que pour une erreur dès la détection.

### 2.4.3 Campagnes d'évaluation des entités nommées

Après avoir présenté les types les plus courants d'entités nommées, la manière dont sont constitués les corpus et les métriques qui permettent d'évaluer les approches pour la reconnaissance d'entités nommées, nous reportons en table 2.1 une liste des principales campagnes d'évaluations conduites sur cette problématique, par ordre chronologique.

| **Date** | **Campagne** | **Langue, modalité** | **Types** | **Métriques** |
|---|---|---|---|---|
| 1996 | MUC-6 | anglais écrit, rapports | pers, org, loc | f-mesure |
| 1997 | MUC-7 | anglais écrit, journalistique | pers, org, loc, date, heure, montant, pourcent | f-mesure |
| 1997 | MET-1 | espagnol, chinois et japonais, écrit journalistique | pers, org, loc, date, heure, montant, pourcent | f-mesure |
| 1998 | MET-2 | chinois et japonais, écrit journalistique | pers, org, loc, date, heure, montant, pourcent | f-mesure |
| 1999 | IREX | japonais, écrit journalistique | pers, org, loc, artefact, date, heure, montant, pourcent | f-mesure |
| 2002 | CoNLL-2002 | espagnol et flamand, écrit journalistique | pers, org, loc, misc | f-mesure |
| 2003 | CoNLL-2003 | anglais et allemand, écrit journalistique | pers, org, loc, misc | f-mesure |
| 2006 | HAREM | portugais, écrit journalistique | pers, org, loc, temps, œuvre, événement, abstraction, chose, valeur, autre | pondération d'erreurs |
| 2006 | SIGHAN | chinois, écrit | pers, org, loc, entité géo-politique | f-mesure |
| 2007 | ACE07 | anglais, arabe et chinois, écrit journalistique et conversationnel | pers, org, loc, bâtiments, entité géo-politique, armes, véhicules | pondération d'erreurs |
| 2007 | EVALITA 2007 | italien, écrit journalistique | pers, org, loc, entité géo-politique | f-mesure |
| 2008 | ACE08 | anglais et arabe, écrit journalistique et conversationnel | pers, org, loc, bâtiments, entité géo-politique | pondération d'erreurs |
| 2009 | ESTER2 | français, oral journalistique | pers, org, loc, temps, montant, fonction, produit | SER |
| 2011 | EVALITA 2011 | italien, oral journalistique | pers, org, loc, entité géo-politique | f-mesure |
| 2012 | ETAPE | français, oral journalistique et conversationnel | pers, org, loc, temps, montant, fonction, produit | SER |

TABLE 2.1 – Caractéristiques des principales campagnes d'évaluation

## 2.5 Proposition de définition des entités nommées

Nous l'avons vu en section 2.2.1, diverses théories linguistiques cherchent à faire la part des choses entre, d'une part, les expressions qui désignent de manière rigide des objets mentaux (noms propres et descriptions définies), et d'autre part, le sens associé à des énoncés

ici restreint aux représentations logiques de propositions sur lesquelles un raisonnement peut s'appuyer. L'historique des campagnes en recherche d'information nous permet de faire un rapprochement avec les objets que l'on cherche à localiser dans des documents et à mettre à la disposition d'applications, comme vu en section 2.2.2, qui ont été appelés *entités nommées*. Enfin, les typologies examinées en section 2.3, les processus d'annotation et d'évaluation décrits en section 2.4 nous paraissent éclairantes à la fois pour mieux définir ces expressions qui désignent et pour élaborer des procédures qui reconnaissent systématiquement les éléments sollicités par la recherche d'information.

Dans ce contexte, l'hypothèse que nous sommes amenés à formuler et que les entités nommées, lorsqu'elles sont résolues, semblent désigner des objets mentaux de manière *stable*, à partir desquels il est attendu qu'une représentation logique *opère*. Ces deux propriétés nous semblent définir les entités nommées, dès lors que l'on ne les décrit ni en extension, ni par le prisme des applications qui les sollicitent. De manière schématique, nous faisons l'hypothèse qu'entre le monde du langage et celui des représentations mentales, la reconnaissance des entités nommées est une interface qui associe des référents à des expressions linguistiques, avant de déterminer les relations logiques sur ou entre ces éléments donnent sens aux énoncés. Plus précisément, voici le formulation de ces deux propriétés que nous proposons d'associer aux entités nommées *résolues* :

- **Stabilité** : une entité nommée résolue désigne de manière rigide un référent, cette désignation n'évolue pas au long de l'énonciation et ne résulte pas d'inférences logiques.
- **Opérabilité** : les entités nommées résolues ne peuvent à elles seules former des propositions et ont vocation à prendre part à des opérations logiques (prédication, quantification, etc.)

Ces deux propriétés, à défaut d'associer un *sens* aux entités nommées, leur permettent tout de même d'être élaborées selon des procédés qui ne relèveront pas de la logique. En conséquence, il est possible de construire une entité nommée à l'aide de ce que nous appellerons des *instructions*. L'élaboration de ces objets est plus systématique : soit elles désignent un objet mental par accès à une connaissance en mémoire (*'Paris'*, *'le Général de Gaulle'*), soit la construction de l'objet mental résulte d'un calcul plutôt que d'un raisonnement (*'le 18 novembre 2012'*, *'5, avenue de la République'*, *'le directeur du département d'informatique'*, etc.).

Ainsi, en nous focalisant sur la désignation et sur la description de réalités linguistiques recouvertes par les entités nommées, nous circonscrivons le champ des expressions que nous envisageons de reconnaître. Par suite dans le cadre de l'automatisation de la reconnaissance des entités nommées, l'approche que nous proposons pourra s'appuyer sur une dynamique d'*instructions locales* supposée sous-jacente aux entités nommées. La diversité des expressions linguistiques considérées nécessite une certaine exhaustivité. Mais, de notre point de vue, il n'apparaît pas nécessaire, dans l'immédiat, de mettre en œuvre des approches logiques pour les reconnaître. Les vues que nous exposons ici nous permettent de prendre position à l'égard des approches existantes et de celles que nous serons en mesure de proposer.

# Chapitre 3

# Approches pour la reconnaissance d'entités nommées

Rappelons ici que notre travail se focalise sur les systèmes informatiques réalisant la *détection* et la *reconnaissance* automatique des entités nommées : nous ne nous occupons pas de leur *résolution*.

Nous utilisons des outils et ressources classiques en TAL (décrits en section 1.3) pour fabriquer un système visant à reconnaître automatiquement les entités nommées. Comme exposé précédemment, pour les ressources, nous utilisons en particulier des lexiques, des transducteurs et des corpus. Pour ce qui concerne les outils, les textes pris en considérations seront pré-traités en morphologique et en morpho-syntaxe (prétraitements qui peuvent potentiellement également faire appel aux mêmes ou à d'autres ressources).

De nombreux systèmes exploitent ces mêmes outils et ressources afin de reconnaître les entités nommées. L'objectif étant évidemment de concevoir des systèmes de bonne *qualité* (critère lié à la *précision*), suffisamment exhaustifs (critère relatif au *rappel*) et robustes (tolérance au *bruit*). Ces trois exigences étant difficilement atteintes simultanément, il s'agit généralement de trouver le meilleur compromis possible lors de l'élaboration et du paramétrage de ces systèmes.

## 3.1 Les approches orientées connaissances

Historiquement, les premières approches (qui se concentraient sur les noms propres) ont focalisé sur la détermination des critères morphologiques (majuscules) pour *détecter* ces entités. Mais rapidement, ces indices étant insuffisants pour *reconnaître* les éléments d'intérêt, les travaux se sont tournés vers une approche plus pragmatique : la constitution de listes exhaustives d'entités nommées, les lexiques du système. Ceux-ci permettent d'atteindre une grande précision, particulièrement lorsque l'ajout d'entrées dans ces listes est contrôlée (notamment en évitant les éléments ambigus). Remarquons à ce sujet que l'utilisation de lexiques peut être directement formalisée comme la reconnaissance d'un langage : la théorie des automates (c.f. 1.3) s'y prête tout particulièrement. De nombreux travaux ont porté sur la conversion automatique de lexiques (ou dictionnaires) sous forme

d'automates : dans le cas général, cette problématique est aujourd'hui bien maîtrisée.

Inévitablement, ces lexiques ne sont pas exhaustifs et certaines entités nommées peuvent ne pas y être présentes dans leur forme exacte. Les noms propres, par exemple, appartiennent à une classe réputée *ouverte* : il n'est pas possible d'en établir une liste finie car de nouveaux noms propres se créent continuellement. De plus, tenir à jour un lexique exhaustif de toutes les entités nommées dans leur forme exacte est laborieux et peu efficace. Les systèmes se sont alors tournés vers la prise en compte d'indices contextuels, qui peuvent utiliser les éléments morphologiques, morpho-syntaxiques ou lexicaux des entités (preuves internes) ou de leurs contextes (preuves externes) [McDonald, 1996, Friburger, 2002]. Voici quelques illustrations de preuves internes :

- **Noms propres** : le mot commence par une majuscule (*'Pompidou'*)
- **Personnes** : le premier token appartient est un prénom (*'Georges Pompidou'*)
- **Dates** : le premier et le dernier token sont composés de chiffres (*'5 juillet 2012'*)
- **Organisations** : le dernier token est *"S.A."* ou *"SARL"* (*'Eiffage S.A.'*)
- **Lieux** : contient *"sur"* ou *"en"* suivi d'un nom de cours d'eau (*'Montlouis-sur-Loire'*)
- ...

Signalons immédiatement qu'il n'a pas été établi à ce jour d'isomorphisme entre un ensemble de propriétés et une classe d'entités nommées : aucun de ces indices, aucune de leurs combinaisons ne s'est montrée à la fois nécessaire et suffisante pour reconnaître un type d'entité nommée. Cependant, ces indices sont à ce jour indispensables lorsqu'il s'agit reconnaître des entités nommées qui ne sont pas présentes dans les lexiques. De plus, comme pour les lexiques, la recherche de ces indices peut être efficacement réalisée à l'aide d'automates.

La généralisation de ces techniques (et leur utilisation dans d'autres domaines) ont conduit à l'implémentation de boîtes à outils [1] qui facilitent la conception de ces automates tout en reposant sur les mêmes exigences en terme de contrôle. Les interfaces, en mode graphique, permettent d'implémenter des *grammaires locales* aisément, chaque règle de reconnaissance étant associée à un diagramme d'états (ou une représentation approchante) dans lequel les arêtes et les nœuds représentent les transitions et états de l'automate. Les transitions sont réalisées par la présence d'indices morphologiques, morpho-syntaxiques ou lexicaux.

Par extension, ces traitements visant à produire une *annotation* (c.f. 2.2) (`'<pers>'`, `'</pers>'`, `'<org>'`, `'</org>'`, ...), il est utile d'indiquer au sein de l'automate quelle modification apporter au texte dès lors qu'une expression a été reconnue. Ceci est réalisé à l'aide de transducteurs, aujourd'hui couramment utilisés pour ce type de tâche [Favre *et al.*, 2005, Stern et Sagot, 2010, Brun et Ehrmann, 2010, Béchet *et al.*, 2011, Maurel *et al.*, 2011].

La figure 3.1 illustre un transducteur complexe, destiné à reconnaître un parti politique. La reconnaissance ne sera possible que pour une expressions linguistique correspondant à un chemin depuis le nœud initial (triangle le plus à gauche) vers le nœud final (carré à droite). Divers items lexicaux sont précisés dans le transducteurs (*'parti'*, *'groupe'*, *'ligue'*), d'autres font appel à des lexiques externes (*'Sigle'*, *'Organisation'*). Certains éléments sont contraints selon un étiquetage morpho-syntaxique (*'A'* pour adjectif, *'N'* pour nom). Les

1. Par exemple Unitex : `http://www-igm.univ-mlv.fr/~unitex/`

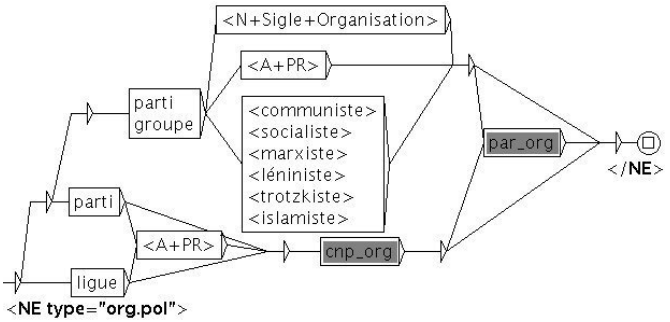



FIGURE 3.1 – Transducteur reconnaissant un parti politique

boîtes, *'cnp_org'* et *'par_org'* représentent l'inclusion de sous-automates (factorisations au sein de la base d'automates). Enfin, *'<NE type="org.pol">'* et *'</NE>'* précisent les balises à insérer lorsque le transducteur reconnaîtra une entité.

Ces transducteurs peuvent être organisés sous forme de *cascades* [Brun et Ehrmann, 2010, Maurel *et al.*, 2011]. Comme un transducteur est en mesure d'insérer des balises au sein des données, d'autres transducteurs peuvent explicitement tirer parti d'unités préalablement reconnues. Sur un mode incrémental, les transducteurs peuvent donc être utilisés en série. Dans ce cas, l'ordre dans lequel sont appliqués les transducteurs peut revêtir une importance capitale : certaines entités peuvent contenir d'autres entités (par exemple *'Le secrétaire général du parti socialiste'*) qui devront alors être détectées préalablement.

Dans ce mécanisme, l'application de transducteurs n'est pas totalement déterministe : pour un texte et un transducteur donné, de très nombreuses annotations peuvent résulter de l'utilisation des transducteurs pour reconnaître des entités. Ces cas relèvent d'*ambiguïtés* (plusieurs reconnaissances possibles) qui nécessitent de prendre une décision. Nous n'entrons pas dans la description détaillée des cas possibles, mais indiquons simplement que dans les systèmes que nous utilisons, parmi les solutions possibles, l'occurrence la plus à gauche et la plus longue sera retenue.

En conclusion, si pour ces approches de nombreuses variantes peuvent être envisagées, le fait de reposer sur un formalisme bien connu dans ses fondements théoriques et très contrôlé en pratique donne une grande importance aux lexiques (par leur contenu) et aux transducteurs (par le langage qu'ils reconnaissent). Traditionnellement, ce lexique et les transducteurs sont implémentés manuellement ou semi-automatiquement (les lexiques sont généralement au moins *validés* manuellement) : ces ressources résultent donc d'un transfert très contrôlé de la connaissance humaine vers le modèle, voilà pourquoi nous les appelons *approches orientées connaissances.*

## 3.2 Les approches orientées données

### 3.2.1 Prendre en compte des indices locaux

Dans les années 90, d'autres approches ont émergé suite à la disponibilité d'importants volumes de données pour des problématiques identifiées. Or, pour des besoins applica-

tifs ou d'évaluation, certains jeux de données ont été *qualifiés* manuellement ou semi-automatiquement. Pour une tâche spécifique, cette qualification peut relever d'une simple classification binaire (*détection* : le phénomène est-il observé dans les données) jusqu'à un travail plus approfondi (*reconnaissance* ou *résolution*) comme par exemple une structuration en profondeur (format XML, enregistrements d'une base de données) ou la découverte de relations entre documents (liens, pointeurs entre documents ou vers un référentiel).

Bien entendu, les tâches pour lesquelles les systèmes sont capables de réaliser des traitements automatiquement sans que le risque d'erreur soit jugé trop important (calculs, envoi de courriels, déclenchement d'opérations comptables, enregistrement / manipulation / diffusion de contenus multimédias, correction orthographique) ont été outillées par l'implémentation directe des procédés sous forme d'automates. Cependant, pour de nombreuses autres tâches, l'intuition conduisait à penser qu'elles pouvaient être résolues automatiquement, à l'aide de modèles plus abstraits, plus complexes, nécessitant une implémentation et un paramétrage.

Ainsi, disposant pour certaines tâches de nombreux exemples des données entrées dans leur forme brute (numérisée) et des données attendues en sortie (qualifiées), il devient possible d'examiner systématiquement les correspondances d'une représentation vers l'autre. Dans ce contexte, concevoir le système peut se focaliser sur l'élaboration d'un modèle *paramétrable* qui vise à transformer la première représentation en la seconde (quelles données en entrée sont potentiellement pertinentes, quelles règles de transformation sont à disposition, comment tenir compte des erreurs pour ajuster les paramètres, etc.). Une procédure automatique et itérative, dite d'*apprentissage automatique* [Mitchell, 1997], sera alors chargée d'ajuster les paramètres disponibles, cette procédure étant guidée à chaque itération par les erreurs que commet le système sur les jeux de données disponibles.

Dans notre cas, la représentation source est le texte brut (sans entités nommées), la représentation cible contient les annotations en entités nommées (c.f. 2.2). Mais les algorithmes d'apprentissage existants ont été majoritairement tournés vers des tâches de classification (binaire ou multi-valuées) : détection de pourriels, transcription d'images (glyphes) en textes (symboles d'un alphabet), répartition de documents au sein de thématiques, etc. Dans ce cadre, l'apprentissage automatique se rapproche plutôt d'un étiquetage (attribution d'une étiquette à un élément) que d'une annotation (délimitation d'une expression).

Prenons pour exemple l'énoncé suivant, annoté en entités nommées :

```
'En <date> 1969 </date> <pers> Georges Pompidou </pers> dirige la
<org> France </org>'
```

En première approche, le texte doit être converti afin que chaque unité atomique (token) reçoive une classe (type d'entité nommée). Pour cela, et afin de pouvoir déterminer les frontières (même lorsque deux entités de même type sont contiguës), le format *BIO* s'est imposé. Au premier token d'une entité de type $t$ sera affectée la classe `'B-t'` (*Begin*). Si l'entité contient plusieurs tokens, les tokens suivants seront classifiés `'I-t'` (*Inside*). Enfin, les mots qui ne sont partie d'aucune entité recevront la classe `'O'` (*Outside*). La figure 3.2 illustre ceci pour notre exemple.

Cette représentation par classes associées aux mots permet d'utiliser des modèles qui estiment la probabilité des classes pour les tokens du texte. Pour ce faire, des statistiques peuvent être recueillies sur les données exemples. En première approximation, nous pouvons

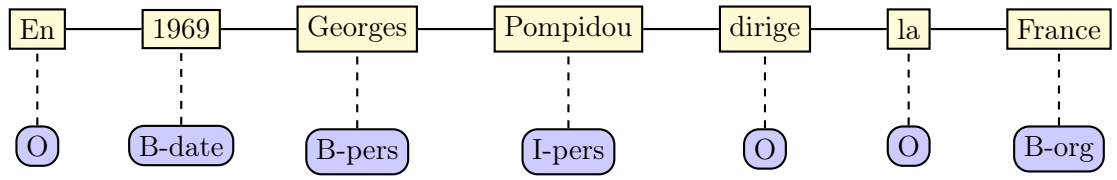



FIGURE 3.2 – Représentation BIO d'une annotation

simplement affecter à chaque token la classe qui lui est *majoritaire* (la plus fréquente) dans le corpus d'apprentissage. Pour ceci, nous considérons les données d'apprentissage comme un ensemble de paires $Ex = \{(t_i, c_i)\}$ : les tokens et les classes qui leur sont associées. Pour annoter un token $t$, le système sélectionne la classe $c$ qui maximise la fréquence $|\{(t_i, c_i) \in Ex, t_i = t, c_i = c\}|$.

Immédiatement, nous voyons que ceci se heurte à deux difficultés majeures :

- le token peut ne pas être présent dans le corpus d'apprentissage (notamment pour les noms propres, catégorie *ouverte*),
- les mots ambigus (*'Washington'* peut-être une personne, un lieu ou une organisation) présenteront systématiquement un taux d'erreur directement lié à leur degré d'ambiguïté.

L'objectif devient alors de tirer parti d'une information plus riche à propos des tokens, généralement issues des ressources (dont les lexiques). Nous pouvons reformuler le modèle majoritaire de manière équivalente pour n'importe quelle information $f(t)$ qui concerne le token par estimation d'une *probabilité conditionnelle* :

$$P(c|f(t)) = \frac{|\{(t_i, c_i) \in Ex, f(t_i) = f(t), c_i = c\}|}{|\{(t_i, c_i) \in Ex, f(t_i) = f(t)\}|}$$

Par suite, il importe d'être en mesure de tenir compte conjointement de plusieurs informations (morphologiques, morpho-syntaxiques, lexicales) à propos du token comme autant de *fonctions caractéristiques*. Ces indices, s'ils sont pertinents, interviennent lors de l'estimation des probabilités des classes. La figure 3.3 illustre ceci.

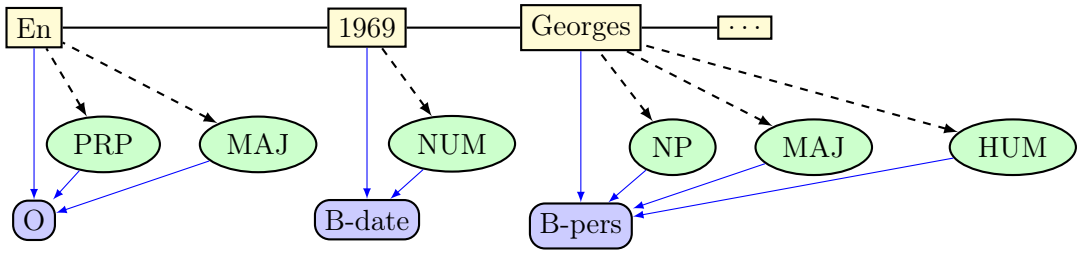



FIGURE 3.3 – Prendre en compte les caractéristiques des tokens

Nous ne faisons pas ici le détail des modèles numériques adaptés à ce type de problème. Parmi celles-ci figurent les modèles bayésiens, le clustering [Miller *et al.*, 2004], le

maximum d'entropie (*MaxEnt* (ou régression logistique *logit*), les machines à vecteur de support [Isozaki et Kazawa, 2002] (*SVM*) et bien d'autres. Pour nos besoins, nous rappelons tout de même la formule du maximum d'entropie qui a démontré son efficacité pour la reconnaissance d'entités nommées [Borthwick *et al.*, 1998, Mikheev *et al.*, 1999, Ekbala *et al.*, 2010], mais aussi pour de nombreuses autres tâches. Par ailleurs, ces modèles sont particulièrement adaptés à la prise en compte de multiples traits discriminants, qui peuvent être interdépendants.

Pour le maximum d'entropie, la probabilité pour un token $t$ d'appartenir à une classe $c$ selon un ensemble de fonctions caractéristiques (indices) $f_1 \dots f_k$ et leurs *poids* associés $\lambda_1 \dots \lambda_k$ (avec $Z(t)$ un facteur de normalisation sur le token) est estimée selon la formule [Berger *et al.*, 1996] :

$$P_\lambda(c|t) = \frac{exp\left(\sum_{i=1}^{n} \lambda_i * f_i(t,c)\right)}{Z(t)}$$

Les diverses ressources (morphologiques, morpho-syntaxiques, lexicales) seront donc exploitées par ces modèles sous la forme de fonctions caractéristiques (ou *traits*, en anglais *features*). La théorie de l'apprentissage automatique fournit une procédure qui ajuste itérativement les poids $\lambda_i$ selon les exemples (et fonctions caractéristiques) mal classifiés par le modèle. Cela est couramment à l'aide d'algorithmes de *descente de gradient*, qui modifient progressivement les poids afin de parvenir à un optimum local.

Finalement, le texte à traiter étant formé d'une séquence de tokens $\langle t_1, t_2 \dots t_n \rangle$, il convient, pour une séquence de classes $\langle c_1, c_2 \dots c_n \rangle$ qui peuvent y être affectés, d'une part de vérifier que ces classes forment une annotation valide (en format *BIO*, une classe *I-t* ne peut être précédée que par un *I-t* ou un *B-t* de même type) et d'autre part d'estimer la *vraisemblance* de l'annotation produite. En pratique, une hypothèse d'indépendance entre classes successives est couramment faite, la vraisemblance de la séquence de classes selon les probabilités locales devenant alors simplement :

$$P_{ME}(\langle c_1 \dots c_n \rangle | \langle t_1 \dots t_n \rangle) = \prod_{i=1}^{n} P(c_i|t_i)$$

Ce modèle tire avantageusement parti des multiples indices, potentiellement interdépendants, qui peuvent être relevés pour un token donné. Cependant il présente un inconvénient majeur : l'estimation des probabilités des classes pour un token ne dépend pas des estimations sur les tokens le précédant ou lui succédant. En effet, pour l'exemple *'Georges Pompidou'*, il est alors possible qu'un modèle à maximum d'entropie, qui examine le token courant sans tenir compte du contexte immédiat, classifie *'Georges'* comme personne, puis *'Pompidou'* comme une organisation (abréviation du *'Centre Pompidou'*).

### 3.2.2 Tirer parti de la séquentialité

La difficulté rencontrée par la classification locale peut-être partiellement résolue en utilisant des modèles tenant compte des dépendances entre classes successives. A cet effet,

les Modèles de Markov Cachés (en anglais *HMM*, *Hidden Markov Model*) sont bien adaptés [Bikel *et al.*, 1999]. Les classes associées aux tokens sont alors interprétés comme *états* du modèle. Dits *génératifs*, l'objectif est alors d'estimer la probabilité qu'une séquence d'état corresponde aux (ou *génère* les) tokens observés. Ce modèle tire également parti des statistiques issues des exemples : l'hypothèse est faite qu'un modèle d'états existe (dans notre cas correspondant aux classes d'entité nommées) et que celui-ci a conditionné le corpus d'apprentissage et conditionnera pareillement les textes à traiter.

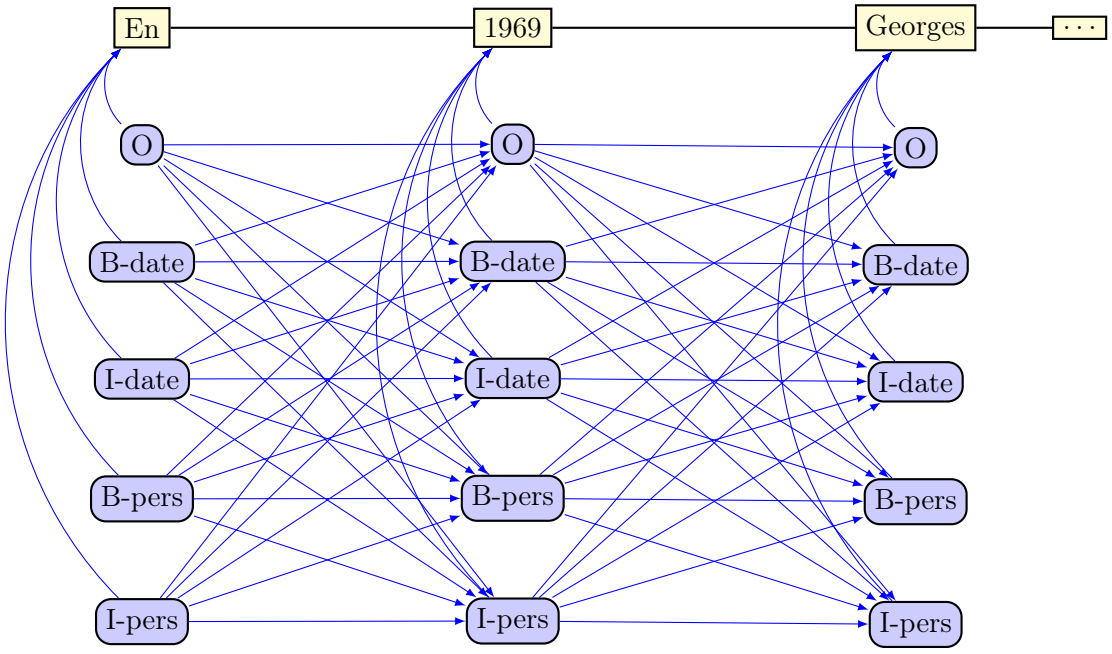



FIGURE 3.4 – Modèle de Markov Caché

La figure 3.4 illustre les dépendances prises en compte par le *HMM* : à la fois celles concernant les transitions d'un état vers un autre état (au sein de la séquence) et celles de génération des tokens par les états. Plus précisément, le calcul de la probabilité de génération d'une séquence tient compte de :

- **Probabilités initiales**, $P(c_1)$ concernant le premier état d'une séquence,
- **Probabilités de génération**, $P(t_i|c_i)$ que le token soit généré par un état donné,
- **Probabilités de transition** $P(c_i|c_{i-1})$ entre deux états successifs.

Ainsi, lorsque le HMM a été paramétré sur le corpus d'apprentissage, il devient possible d'exploiter les dépendances entre états successifs en calculant la probabilité d'une séquence d'états $\langle c_1, c_2 \dots c_n \rangle$ conjointement à une séquence de tokens $\langle t_1, t_2 \dots t_n \rangle$ (avec $P(c_1|c_0) = P(c_1)$) selon la formule :

$$P_{HMM}(\langle c_1 \dots c_n \rangle, \langle t_1 \dots t_n \rangle) = \prod_{i=1}^{n} P(c_i|c_{i-1}) * P(t_i|c_i)$$

Les paramètres du modèles peuvent être initialisés par calcul direct de probabilités

conditionnelles sur le corpus, puis éventuellement affinées selon diverses approches (Baum-Welch, EM, etc.). De la même manière que précédemment, une fois paramétré, des algorithmes de programmation dynamique (*Viterbi*) permettent de déterminer la séquence d'état la plus probable étant donné les tokens rencontrés. L'inconvénient majeur de cette approche est de ne pouvoir, dans sa version initiale, tenir compte des multiples indices dont on dispose pour un token donné.

Nous remarquons que ces deux modèles exploitent respectivement chacun des deux axes que nous avons évoqués au début de ce document (c.f. 1.1) : l'*ontologie* pour le maximum d'entropie (enrichissement d'un token par de multiples ressources morphologiques, morpho-syntaxiques ou lexicales) et la *structure* pour les modèles markoviens (dépendance entre des annotations contiguës). Il s'ensuit que de nombreuses tentatives ont été menées pour concilier ces deux approches au sein d'un modèle unifié. Ont alors émergé les modèles markoviens à maximum d'entropie [McCallum *et al.*, 2000] (*MEMM*, Maximum Entropy Markov Models), puis assez rapidement les champs aléatoires conditionnels [Lafferty *et al.*, 2001] (*CRF* en anglais, *Conditional Random Fields*).

Les champs aléatoires conditionnels sont une évolution des *HMM* en tenant compte de multiples *traits discriminants* conditionnant (comme pour le *maximum d'entropie*) les états d'une séquence. Ce sont des modèles dits *graphiques* car ils opèrent sur des cliques d'états et de fonctions caractéristiques. Ainsi les probabilités conditionnelles ne portent plus sur une seule classe, mais sur des ensembles d'états dépendants les uns des autres. Dans notre cas, les classes à reconnaître formant des séquences, les dépendances *markoviennes* entre états sont linéaires et les cliques contiendront généralement des *bigrammes* de classes, comme illustré en figure 3.5.

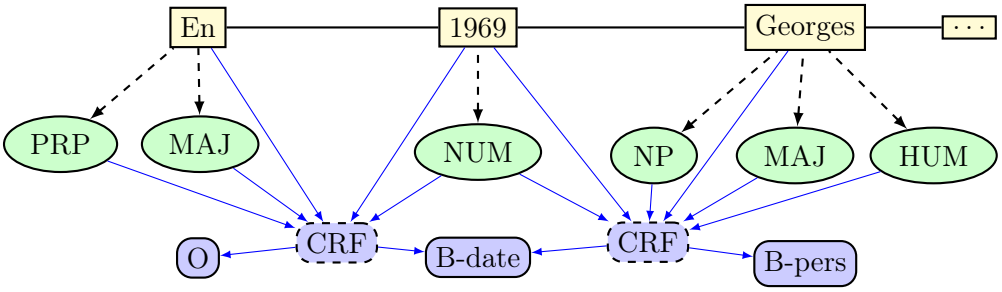



FIGURE 3.5 – Modèle graphique CRF

Ajoutons de surcroît, sans entrer dans les détails, que les poids sont ajustés sur des séquences entières, ce qui évite de normaliser les probabilités au niveau du token et contourne le *"label bias problem"* (effet de bord de la normalisation des masses de probabilités sortante d'un état). La formulation porte donc sur la probabilité d'une séquence de classes $\langle c_1, c_2 \dots c_n \rangle$ pour une séquence de tokens $\langle t_1, t_2 \dots t_n \rangle$, dès lors que le modèle tient compte des caractéristiques $f_1 \dots f_k$ et de leurs *poids* associés $\lambda_1 \dots \lambda_k$ (avec $Z(t)$ un facteur de normalisation sur la séquence) :

$$P_{CRF}(\langle c_1 \dots c_n \rangle, \langle t_1 \dots t_n \rangle) = \frac{exp\left(\sum_{i=1}^{n}\sum_{j=1}^{k} \lambda_j f_j(c_{i-1}, c_i, t_i)\right)}{Z(t)}$$

Leur bonnes performances et leur robustesse pour la reconnaissance des entités nommées (mais également pour d'autre tâches, comme l'étiquetage morpho-syntaxique) en font aujourd'hui le modèle réputé le plus adéquat à ce jour [McCallum et Li, 2003, Raymond et Fayolle, 2010, Savary *et al.*, 2010, Zidouni *et al.*, 2009]. Une critique courante est cependant qu'ils restent difficiles à interpréter et à capitaliser : même lorsque l'on peut extraire les traits les plus discriminants, ceux-ci sont généralement composites et exhibent des dépendances complexes dont il est diffile d'affirmer qu'elles sont nécessaires ou suffisantes pour déterminer les entités nommées.

Ces trois approches (*maximum d'entropie*, *HMM*, *CRF*) se basent sur une même représentation *plate* des entités nommées. Cependant les derniers travaux sur le sujet ont mis en avant la nécessité de reconnaître à minima des *structures* imbriquées plutôt que de simples *classes* ou *étiquettes*. Ceci semble cohérent avec la tâche initiale, qui relevait plutôt d'une annotation, mais a tiré parti de modèles existants orientés vers la classification (à l'aide des étiquettes *BIO*). Les mêmes modèles, qui *aplatissent* la structure, modélisent séparément chaque niveau ou utilisent des grammaires probabilistes (*PCFG*) pour déterminer la structure, ont été appliquées avec un certain succès [Finkel et Manning, 2005, Dinarelli et Rosset, 2011].

En conclusion, nous voyons que les entités nommées ont été simultanément considérés comme information nécessitant des indices à la fois en profondeur (*ontologiques*, liés aux ressources) et en largeur (*structurels*, liés au contexte). Les approches que nous avons exposées ici se focalisent sur des modèles qui se paramètrent automatiquement. S'ils ne sont pas dépourvus d'un travail préalable conséquent (préparation des jeux de données, implémentation du modèle, des procédures d'apprentissage et d'estimation, sélection des traits et dépendances pertinents, etc.), ils sont supposés s'instancier dès lors que des données leur permettent d'ajuster leurs paramètres sur une tâche particulière, voilà pourquoi nous les appelons *approches orientées données*.

Signalons également que, si les approches orientées données permettant un paramétrage automatique selon de multiples indices, rien n'empêche alors de faire provenir ces fonctions caractéristiques de la sortie d'autres systèmes. De nombreux travaux ont été menés en ce sens, nous n'en décrivons pas le détail (chaque approche étant elle-même complexe, puisque combinaison de sous-systèmes), mais relevons que données et connaissances se trouvent alors mêlées [Borthwick *et al.*, 1998, Brun *et al.*, 2009b, Brun *et al.*, 2009a, Zidouni *et al.*, 2010, Nouvel *et al.*, 2012], ce qui a pour avantage de permettre une grande richesse de traits discriminants, mais a l'inconvénient de nécessiter un paramétrage lié à la configuration de chaque sous-système, susceptible de nécessiter un apprentissage à chaque fois qu'un sous-système est modifié.

## 3.3 Proposition d'approche : les marqueurs d'annotation

Nous ne chercherons pas à discuter ici des problématiques de temps humain consacré à élaborer tel ou tel type de système qui nous paraît conséquent dans tous les cas, même s'il peut-être mutualisé (campagnes d'évaluation). Nous essayerons par ailleurs de rapprocher la définition théorique des entités nommées (c.f. 2.5) des possibilités que nous offrent les approches orientées connaissances et données. Nous avons à disposition des données, des systèmes orientés données ou connaissances, que nous chercherons à mettre au service d'une problématique d'*annotation*.

De notre point de vue, le grand avantage des systèmes orientés connaissances réside dans la *structure* qu'ils apportent au modèle. Leur élaboration incrémentale nécessite des étapes de factorisation de divers éléments liés à la reconnaissance d'entités nommées. Cela permet à ces systèmes de retracer de manière précise les informations dont il a été fait usage et les constructions qui ont été incrémentalement reconnues.

En ce qui concerne les entités nommées, cette notion étant pour partie artificielle, de nombreux travaux peuvent encore être nécessaires avant d'obtenir un consensus sur leur nature. Dans ce cadre, les méthodes orientées données nous semblent permettre l'évaluation objective de l'apport des analyses et des contextes pour leur reconnaissance. Ainsi, selon le modèle implémenté, il devient possible de valider (ou d'invalider) des hypothèses à propos de l'objet d'étude.

Dans notre travail, nous formulons les hypothèses suivantes :

- Les *entités nommées*, de par leur nature *stable*, sont des éléments locaux d'un énoncé et ne contiennent pas de relations internes.
- Les *entités nommées*, afin d'être *opérables*, disposent d'une structure interne fortement contrainte par contiguïté, qui peut être mise en correspondance avec des *instructions* sous-jacentes.
- Une *instruction* peut-être assimilée à une transition qui réalise un changement d'état lors de la construction d'un objet mental.
- Les *instructions* sont des éléments abstraits qui ne peuvent être associées à un token en particulier, mais sont plutôt déterminées par la présence d'indices s'appuyant simultanément sur l'*ontologie* et la *structure* du contexte local.

Dans ce contexte, nous prenons de la distance, comme annoncé précédemment, avec les approches orientées données, ceci tout particulièrement avec l'hypothèse que la reconnaissance d'entités nommées peut-être formulée comme l'attribution de classes aux tokens d'un texte. Cependant, nous ne nous positionnons pas non plus dans la ligne des approches traditionnelles orientées connaissances, qui sont généralement d'une nature globale et totalement contrôlée, ce qui les rend généralement moins robustes.

De fait, nous observons la représentation réalisée par les humains (et leurs connaissances) sur les données : l'*annotation*. Et nous considérons ainsi que chaque balise, que nous nommons **marqueur d'annotation** (‘<pers>’, ‘</pers>’, ‘<org>’, ‘</org>’, ‘<loc>’, ‘<date>’, ‘<func>’, etc.) est une instruction qui participe à la structuration du texte en entités nommées. Si nous reprenons l'exemple précédent, et y intégrons ces informations de structure, nous obtiendrons :

‘En <date> <num> 1969 </num> </date>

`<pers> <prenom> Georges </prenom> <famille> Pompidou </famille> </pers>`
`dirige la <org> <loc> France </loc> </org>'`

Notre hypothèse est que, par exemple, `'En <date> <num> 1969 </num> </date>'` contient effectivement quatre instructions distinctes (une par marqueur) et que chaque instruction est conditionnée par des indices issus des éléments disponibles en ontologie (ici la morphologie de *'1969'* comme nombre partie d'une date), en structure (ici le token précédent *'En'*), ou par combinaison des deux (la catégorie morpho-syntaxique du token précédent, ici une préposition). Il s'agira alors d'estimer localement les marqueurs probables, avant de déterminer, par leurs combinaisons, l'annotation la plus vraisemblable. Grâce aux techniques de fouille de données, nous observerons les motifs qui conditionnent l'apparition des marqueurs. Nous serons alors en mesure d'élaborer un système de reconnaissance d'entités nommées selon cette approche originale et d'évaluer ses performances en les comparant aux approches existantes.

Deuxième partie

# Exploration de données pour extraire des règles d'annotation

# Chapitre 4

# Fouille de données textuelles

## 4.1 Généralités sur l'exploration de données

L'exploration (ou fouille) de données englobe un ensemble de techniques ayant pour objectif d'analyser de grands volumes de données [Han et Kamber, 2005]. Ces techniques ont pris leur essor avec la disponibilité croissante de données dans divers domaines d'application. Dès lors que l'automatisation de la collecte des données (capteurs, codes-barres, reconnaissance automatique de textes, généralisation des dispositifs de saisie) a donné lieu à la constitution de grandes bases de données de diverses natures, lorsqu'il s'agit d'analyser ces données, il devient évidemment très coûteux de les examiner une à une, manuellement.

Une tâche d'analyse peut suivre un ojectif précis, lorsque l'on sait quel objet d'étude l'on souhaite modéliser et prédire. Dans d'autres cas, il s'agit de décrire exhaustivement, sans a priori, des *corrélations*, des *associations*, de manière plus générale de faire émerger des *connaissances* à partir des données. Ces besoins ont rendu pertinent la mise au point des méthodes *objectives* qui permettent d'explorer les données, afin d'en extraire des connaissances, dans des délais raisonnables et selon des critères liés à l'*intérêt* que, potentiellement, ces connaissances peuvent présenter.

La fouille de données propose un ensemble de méthodes qui analysent systématiquement les données et vise à extraire, à partir de *faits*, un modèle de connaissances (par exemple par généralisation), des *motifs* (en anglais *patterns*), qui deviendront exploitables par l'humain ou par un système de plus haut niveau. De notre point de vue, nous considérons que la problématique peut-être interprétée comme une *réorganisation* des données, qui met de côté les informations non-pertinentes (bruit) et utilise les données pertinentes pour construire des structures appropriées (motifs).

De manière générale, même s'il n'y a pas réellement consensus sur le sujet, les processus d'exploration des données tiennent compte des éléments suivants :

- **Alphabet des items** : symboles atomiques pour représenter les données.
- **Base de données** : données numérisées et stockées, sous-ensemble du langage des items, généralement segmentées sous forme de multi-ensemble, les *transactions*.
- **Alphabet des motifs** : symboles atomiques pour représenter les motifs (il peut avoir une intersection avec l'alphabet des données).

– **Motifs** : algèbre formée par le langage des motifs muni d'opérateurs (dont généralement une *relation d'ordre* et pour des motifs séquentiels la *concaténation*).
– **Mesures** : opérateurs permettant de quantifier les motifs au regard des données (en particulier, mesures d'*intérêt*).

La fouille de données a été appliquée à la recherche de solutions pour de nombreuses problématiques. Ces méthodes ont prouvé leur efficacité dans des domaines très divers comme par exemple l'achat en librairies [Srikant et Agrawal, 1996], l'analyse de séquences ADN [Zhu *et al.*, 2007], l'analyse de logs web [Wang et Han, 2004] ou encore le traitement du langage. Nous nous focalisons sur ce dernier domaine d'application, dans le cadre de l'élaboration des ressources à utiliser pour la reconnaissance d'entités nommées, dont les lexiques et les transducteurs.

## 4.2 Fouille de documents textuels pour enrichir les lexiques

Les premières utilisation de méthodes de fouille de données pour améliorer la reconnaissance des entités nommées a essentiellement porté sur l'enrichissement de lexiques. Les premières tentatives en ce sens ne disposaient pas de ressources externes structurées (Web, Wikipedia) et ne pouvaient exploiter que des bases de documents purement textuelles. La problématique était alors d'y détecter des *noms propres* afin de les ajouter à des lexiques. Ceci a été expérimenté par utilisation d'algorithmes itératifs (*bootstrapping*) qui extraient alternativement des noms propres et les motifs contextuels corrélés à ces noms propres [Riloff et Jones, 1999, Dredze *et al.*, 2010]. Un tel système est initialisé avec des noms propres courants de la classe à extraire. Puis un algorithme détermine les contextes qui discriminent ces noms propres dans les textes. Ces contextes permettent alors de *détecter* de nouveaux noms propres, qui sont à leur tour utilisés pour extraire de nouveaux contextes, etc. Ainsi, le lexique est enrichi itérativement. Cette approche, faiblement supervisée, est difficile à contrôler (problème de dérive sémantique, en anglais *semantic drift*) : la couverture et la qualité des lexiques obtenus dépend fortement des paramètres (noms propres initiaux, critères d'arrêt).

D'autre approches se sont popularisées, qui s'appuient sur le Web comme une vaste ressource afin d'en extraire des noms propres [Etzioni *et al.*, 2005, Nadeau, 2007, Mooney et Bunescu, 2005, Béchet et Roche, 2010, Roche, 2010, Downey *et al.*, 2007]. Ces méthodes utilisent généralement les moteurs de recherche qui, par indexation des données, fournissent un matériau *partiellement filtré*. Le système interroge l'index, parfois à l'aide de patrons sémantiques simples (par exemple, le motif `‘"la ville de"’`, avec guillemets, est supposé être suivi de noms de villes dans les documents), et récupèrent les pages retournées par le moteur pour y détecter le ou les items lexicaux d'intérêt. Une analogie est parfois faite avec un écosystème biologique [Etzioni *et al.*, 2005], dans lequel des systèmes dits *herbivores* traitent la donnée brute (moteurs de recherche) et d'autres systèmes, dits *carnivores*, manipulent cette donnée déjà partiellement analysée (utilisation de motifs).

Ces systèmes exploitent à la fois les volumes de données disponibles (ce qui justifie une approche de type *fouille*), des patrons sémantiques (pour interroger les moteurs), mais également, de plus en plus, les listes structurées accessibles sur le Web (tableaux, listes, énumérations, liens, etc.). Faiblement supervisées, ils fournissent des résultats très

satisfaisants. Dans cette méthode, la fouille apparaît cependant réalisée en grande partie par le moteur de recherche : la marge de progression est alors fortement dépendante du moteur utilisé et de la forme des motifs qu'il autorise (il est rare que les requêtes puissent formuler des requêtes s'appuyant sur la morpho-syntaxe, la sémantique, etc.). Par le volume de données disponible sur le Web, Ces approches parviennent à exploiter avantageusement une connaissance, in fine, saisie par l'humain (les pages Web partiellement structurées).

Dans la même ordre d'idées, les encyclopédies en ligne (Wikipedia) sont aujourd'hui devenus une source incontournable d'information qu'il est également possible de fouiller. En effet, elles présentent trois caractéristiques qui les rendent très intéressantes à cet effet. Premièrement, elles couvrent aujourd'hui assez exhaustivement de vastes domaines de connaissances, et sont de surcroît très régulièrement mises à jour. En second lieu, elles sont fortement structurées (au sein des pages autant que par liens qualifiés entre pages), ce qui permet d'explorer ces ressources en profondeur. Enfin, elles ont l'avantage d'être validées par un grand nombre d'utilisateurs, donc de résulter en partie de consensus. Ces encyclopédies sont l'objet de nombreux travaux, pour construire des ontologies mais aussi pour en extraire des entrées lexicales ou des contextes discriminants pour les homonymes [Bunescu et Pasca, 2006, Charton et Torres-Moreno, 2009, Charton, 2009].

Nous disposons déjà de ressources lexicales assez couvrantes, constituées semi-automatiquement, que nous décrivons en section 7.2.2. Si nous n'abordons pas dans ce travail les problématiques d'enrichissement de ces ressources, nous présenterons quelques résultats visant à déterminer leur influence pour la reconnaissance des entités nommées. Signalons dès à présent que nous établirons une distinction nette entre la catégorisation sémantique à l'aide de ressources lexicales et l'utilisation de ces catégories sémantiques au sein de motifs. Il nous paraît en effet que les motifs n'ont pas vocation a remplacer les lexiques, mais à s'appuyer sur eux afin de faire abstraction des items lexicaux eux-même.

## 4.3 Extraction automatique de motifs linguistiques

Si la fouille de données s'abstrait de l'automate pour considérer les données, rien ne l'empêche cependant d'explorer les données afin d'y découvrir des structures apparentées à des automates [Hingston, 2002]. Nous retrouvons ici la vue faite en section 1.3 des *transducteurs* comme *ressources* : il est possible, en fouillant les données, d'en extraire des automates, des grammaires et peut-être, ce sera notre hypothèse, des motifs corrélés aux *instructions* sous-jacentes au langage. Cette tâche est nommée *induction* : elle consiste à extraire des connaissances sous forme d'*automates* à partir de données observées. De nombreux travaux ont été réalisés sur l'induction de grammaires qui décrivent (génèrent, reconnaissent) le langage naturel [Parekh et Honavar, 2000, Mendes et Antunes, 2009].

Cependant, les travaux ne permettent pas véritablement, à ce jour, de circonscrire précisément une forme de motifs plus appropriée pour traiter *tout* le langage naturel et *uniquement* le langage naturel. Cet objectif ambitieux a donc laissé place à l'extraction de motifs pour des phénomènes linguistiques particuliers et pour des tâches spécifiques [Besançon *et al.*, 2006, Sun et Grishman, 2010, Ezzat, 2010, Charton *et al.*, 2011]. Les motifs sont généralement formulés à l'aide de ressources que l'on suppose être utiles à la problématique considérée. La forme des motifs peut être assez variable et s'appuyer sur des formalismes

très divers : logiques, séquentiels, hiérarchiques, relationnels, etc.

Pour le langage naturel, la nécessité de tenir compte de la séquentialité des données est souvent incontournable (concaténation de phonèmes, mots, énoncés, prises de paroles, etc.). C'est donc un sous-domaine de la fouille de données auquel il sera fait appel, qui doit examiner des séquences plutôt que des ensembles, ce qui augmente par ailleurs la combinatoire des motifs à considérer. Certains travaux ont exploré la possibilité d'assouplir partiellement cette contrainte lorsqu'il s'agit de prendre en compte un contexte plus large autours des éléments à reconnaître [Plantevit *et al.*, 2009,Cellier et Charnois, 2010,Charnois *et al.*, 2009]. Pour notre problématique, nous postulons que le contexte à prendre en compte est essentiellement local et requérons que tous les éléments d'un motif soient contigus.

Concernant la problématique spécifique des entités nommées, des travaux ont été menés afin d'extraire des motifs se focalisant sur ces objets linguistiques [Kushmerick *et al.*, 1997, Califf et Mooney, 1999, Freitag et Kushmerick, 2000, Girault, 2008]. Généralement, ces approches supervisées extraient, à partir d'un corpus annoté, des motifs généralisant les contextes dans lesquels apparaissent les entités nommées. Ces motifs peuvent être généralisés à l'aide d'indices morphologiques simples exprimés sous forme d'automates. A notre connaissance, ces méthodes se focalisent sur l'extraction de motifs, et non sur leur utilisation automatique pour réaliser la reconnaissance des entités nommées. Ces motifs sont donc supposés déterministes : ceci amène souvent à privilégier des motifs précis et à écarter les motifs qui ne fournissent que des indices partiels. De plus, l'approche que nous avons développée [Nouvel *et al.*, 2010b, Nouvel et Soulet, 2011, Nouvel *et al.*, 2011a] est à notre connaissance la seule à réaliser la fouille de données qui se focalise sur les marqueurs d'annotation.

Comme indiqué en section 1, nous nous situons dans une approche qui s'attache à considérer les annotations, et plus particulièrement les *marqueurs* (balises) qui forment l'annotation (c.f. 3.3). Comme nous le verrons, le cadre théorique que propose la fouille de données est approprié pour décomposer la problématique de reconnaissance des entités nommées comme recherche de motifs qui insèrent des marqueurs dans les textes. Par ailleurs, nous ne nous contraignons pas à extraire des *connaissances* exactes : de simples indices, même lorsqu'ils apportent une information partielle, peuvent être d'*intérêt*. De ce fait, l'ambiguïté naturellement présente dans les ressources (pour grande partie, dans les lexiques) peut-être conservée jusqu'à l'extraction des motifs. La résolution des cas ambigus sera reportée en fin de processus, lorsqu'il s'agira d'utiliser ces motifs pour reconnaître les entités nommées.

# Chapitre 5

# Extraire des règles comme motifs séquentiels hiérarchiques

## 5.1 Les données comme séquences de tokens

Les données sur lesquelles nous travaillons sont, en dernière analyse, enregistrées comme de simples séquences de caractères. Nous faisons abstraction de cela pour considérer des séquences de *tokens* (mots) (c.f. 1.1). La segmentation sous forme de *tokens* et d'*énoncés* ne constituant pas le cœur de notre travail, nous ne discuterons cet aspect que marginalement. Indiquons simplement que cette étape est généralement réalisée par un automate, qui prend en entrée le flux de caractères et y détecte des séparateurs (blancs, ponctuations, etc.). Le flux est alors segmenté de manière non ambiguë en *tokens* , qui seront pour nous les unités minimales des traitements. De la même manière, les séparateurs d'énoncés (point, point d'exclamation, d'interrogation, etc.) sont également reconnus, ce qui segmente le flux de tokens en *énoncés*.

Comme l'ordre et la contiguïté des éléments (*tokens*) au sein des énoncés nous semble fondamental, la fouille de données doit en tenir compte, elle explorera des motifs dits *séquentiels*. Nous nous appuyons pour ce faire sur les travaux de [Fischer *et al.*, 2005] qui présente les approches à base de *chaînes*. Ces dernières sont formées par concaténation d'éléments de l'alphabet. Pour alléger les notions, nous noterons simplement la concaténation par la présence d'indices sur les éléments d'un énoncé ($e_1e_2 \dots e_n$) ou d'un espace blanc lorsque les items sont instanciés (`'Voici quatre tokens .'`).

Formellement, notre alphabet des données $\Sigma_d$ est un ensemble fini de tokens. Nous tenons pour acquis qu'au sein de ces tokens peuvent être systématiquement distingués les *marqueurs* d'annotation (balises), qui en sont un sous-ensemble $\Sigma_m \subset \Sigma_d$. Par concaténation sur $\Sigma_d$, nous formons le langage des données $\mathcal{L}_d$. Un énoncé $E \in \mathcal{L}_d$ de taille $n$ sera donc la concaténation de $n$ éléments $e_1e_2 \dots e_n$. Enfin, la base de données d'énoncés $\mathcal{D}$ est un multi-ensemble (un même énoncé pouvant apparaître plusieurs fois dans les données) de $\mathcal{L}_d$.

En résumé, nous utiliserons les notations suivantes pour les données :

– $\Sigma_d$ : alphabet sur les données (*tokens* et *marqueurs*),

– $\Sigma_m$ : sous-alphabet des *marqueurs* : $\Sigma_m \subset \Sigma_d$,
– $\mathcal{L}_d$ : langage des données généré par concaténation de tokens : $(\Sigma_d)^*$,
– $\mathcal{D}$ : base de données, multi-ensemble des *énoncés* de $\mathcal{L}_d$.

Cette formalisation décrit les données brutes, avant tout traitement. Cependant, un certain nombre d'analyses peuvent être conduites (morpho-syntaxiques, lexicales, etc.) comme pré-traitements, avant de réaliser la fouille de données. Pour notre cadre théorique, nous réduisons ces analyses à des procédés automatiques d'*enrichissement* des données, sans préjuger de la complexité dont ils peuvent relever.

## 5.2 Motifs s'appuyant sur des ressources ambiguës

Les enrichissements consistent donc à exploiter des ressources pour apporter une information supplémentaire aux tokens, qui n'est pas nécessairement déterminisée. Ainsi, l'analyse morpho-syntaxique et les bases lexicales associent une ou plusieurs catégorie(s) à un token. Par ailleurs, nous hiérarchisons la première sous la seconde : une relation d'ordre (c.f. 5.3.3) nous permettra de généraliser un token successivement à une catégorie morpho-syntaxique puis à une catégorie lexicale. Pour un token (par exemple `'Madrid'`), ceci sera dénoté par préfixation et utilisation de la barre oblique `'/'` (par exemple `'VILLE/NP/Madrid'`). Ces procédés d'enrichissements sont ici réduits à une application que nous noterons $R(E), E \in \mathcal{L}_d$.

Pour ce qui nous occupe dans ce mécanisme d'enrichissement, nous remarquons simplement que ce n'est pas un homomorphisme pour la concaténation sur les tokens, puisqu'il tient compte d'informations contextuelles pour catégoriser un token : $R(e_1e_2) \neq R(e_1)R(e_2)$ : l'enrichissement ne peut donc être réalisé par token, et doit l'être au niveau des énoncés. De plus, à un token peuvent être affectées diverses catégories. Nous fournissons un nouveau cadre formel à la base de données $\mathcal{D}_r$ qui résultera de ces enrichissements, sur laquelle nous nous réaliserons la recherche de motifs. Comme $\mathcal{D}$, c'est un multi-ensemble tel que $\mathcal{D}_r = \{R(E), E \in \mathcal{D}\}$.

Considérons de prime abord cette application dans le cas non-ambigu. Par là, nous entendons que l'enrichissement ne produit qu'une possibilité d'analyse par énoncé. Nous aboutirons alors à la construction, pour un énoncé (ou *chaîne de tokens*) $E = e_1e_2 \dots e_n \in \mathcal{L}_d$, d'une *chaîne d'items* $R(e) = i_1i_2 \dots i_n$. Remarquons que ces deux chaînes sont de même taille : chaque item $i_i$ correspond au token $e_i$ qui a été *enrichi*. Nous formons dès lors l'alphabet des données enrichies $\Sigma_r$ (disjoint de $\Sigma_d$) qui résulte des possibilités d'enrichissements à l'aide des ressources morpho-syntaxiques et lexicales.

Considérons par exemple l'énoncé *'Il visite Madrid.'*. La segmentation en token nous donne quatre tokens $E =$`'Il visite Madrid .'` (le point *'.'* étant un token à part). Dans notre cas, le procédé consistera essentiellement à lemmatiser, déterminer la catégorie morpho-syntaxique et les étiquettes lexicales associées aux tokens en contexte, tout en conservant éventuellement le token lui-même. Ce traitement pourra donc être plus ou moins profond selon le token considéré :

$R(E) =$ `'PRO/il/Il VER/visiter/visite VILLE/NP/Madrid PONCT/.'`

Pour cet exemple `'VILLE/NP/Madrid'` est un item (atomique, élément de l'alphabet $\Sigma_r$),

auquel ont été attachées des informations morphologiques (nom propre, `'NP'`) et lexicales (catégorie sémantique de ville, `'VILLE'`). Pour l'item `'VER/visiter/visite'`, nous remarquons que le traitement morpho-syntaxique lui affecte une catégorie *en contexte*, car le token `'visite'` pourrait également être un nom commun dans un autre contexte. Enfin, le point `'PONCT/.'` n'est catégorisé que par morphologie. Nous reviendrons plus en détail sur cet enrichissement, qui sera adapté selon les tokens et leurs contextes, et pourra s'affranchir des *formes* elles-mêmes.

Par ailleurs, comme nous l'avons mentionné précédemment, les traitements peuvent ne pas être déterministe et produire des analyses ambiguës. Or nous ne souhaitons pas prendre de décision a priori et arbitrairement sur les informations à sélectionner. Nous autorisons donc la présence d'ambiguïtés dans les données enrichies, tout en faisant le choix de ne pas la propager aux motifs que nous extrairons. Voilà pourquoi nous représentons formellement cette ambiguïté par utilisation d'un opérateur de disjonction exclusive $\oplus$ entre items. Notons que cette ambiguïté est liée aux enrichissements (et leurs ressources) et porte, dans notre cas, majoritairement sur les items (et non leur concaténation). Ainsi, pour alléger les notations, cet opérateur sera prioritaire sur la concaténation : $ab_1 \oplus b_2 c$ est équivalent à $a(b_1 \oplus b_2)c$.

Pour illustration, si nous considérons l'exemple précédent, mais en remplaçant *'Madrid'* par le token lexicalement ambigu *'Washington'* (ville ou personnage célèbre `'CELEB'`), nous obtenons alors :

$R(E) =$ `'PRO/il/Il VER/visiter/visite VILLE/NP/Washington`$\oplus$`CELEB/NP/Washington'`

Enfin, pour les marqueurs, nous conservons l'alphabet $\Sigma_m$ pour éviter de surcharger les notations, même s'il est plus pratique, dans l'implémentation, de leur affecter une catégorie à part (par exemple `'EN'` : `'EN/<pers>'`, `'EN/</org>'`, `'EN/<org>'`, etc.). Ainsi, le cadre formel pour la fouille de données devient, à l'aide de cette application d'enrichissement :

- $\Sigma_r$ : alphabet sur les *items*, tokens enrichis en contexte,
- $\Sigma_m$ : sous-alphabet des marqueurs : $\Sigma_m \subset \Sigma_r$,
- $\mathcal{L}_r$ : langage généré par disjonction exclusive et concaténation d'items : $(\Sigma_r \cup \{\oplus\})^*$,
- $\mathcal{D}_r$ : base de données enrichie, multi-ensemble des *énoncés* de $\mathcal{L}_r$.

En théorie, la fouille de données peut explorer des structures assez diverses. Pour limiter la combinatoire, nous nous sommes restreints, comme indiqué plus haut, à la disjonction exclusive $\oplus$. Ceci implique que notre langage des motifs ne peut contenir des items portant sur des disjonctions ou des conjonctions de catégories sémantiques. Pour clarifier ce point, supposons par exemple qu'un lieu soit nommé d'après un personnage célèbre, comme *'Centre Georges Pompidou'* : cela pourrait donner lieu à l'exploration de motifs avec conjonctions (ici notées &) du type :

`'BAT/NP/Centre BAT/NP&CELEB/NP BAT/NP&CELEB/NP'`

Dans notre cas, l'exploration de tels motifs paraît trop coûteuse étant donné le nombre de traits sémantiques dont nous disposons, pour un gain trop incertain. De manière plus générale, il y a là une articulation entre, d'une part, la richesse des traitements en amont de la fouille et, d'autre part, les généralisations à examiner lors de l'exploration des données enrichies. Certaines combinatoires sur les items (disjonctions, conjonctions, négations, etc.) peuvent être transférées de l'une à l'autre, selon la granularité que l'on souhaite explorer lors de la fouille de données. Notre choix est donc de ne faire apparaître les ambiguïtés

que dans les données enrichies et de l'exclure des motifs, qui se contenteront d'explorer les complexités de structure (séquences) et d'ontologie (catégories).

Remarquons également qu'une des représentations les plus riches pour associer diverses propriétés aux tokens est l'utilisation de structures de traits. Celles-ci se formulent comme des propriétés auxquelles sont attachées une valeur, une liste de valeur ou une autre structure de traits. Par exemple, à *'Madrid'* pourrait être associée une structure du type `'(MS:NP),(LEX:(LOC:(CAT:VILLE),(SEM:GEO,ADM,SPORT)))'` qui indique que l'analyse morpho-syntaxique `'MS'` lui associe la catégorie nom propre `'NP'`, et l'analyse lexicale `'LEX'` y associe un trait lieu `'LOC'`, celui-ci étant de la catégorie `'CAT'` ville `'VILLE'` et ayant des interprétations sémantiques `'SEM'` comme objet géographique `'GEO'`, administratif `'ADM'` ou encore sportif `'SPORT'`. Extrêmement flexibles, ces structures présentent une richesse de description qui les rend, telles quelles, trop coûteuses à fouiller.

Nous plaçons un curseur dans la recherche d'un juste compromis entre la richesse de description des éléments et la combinatoire induite lors de l'exploration des données. De ce point de vue, nous faisons le choix de mettre de nombreuses informations à disposition de la fouille de données, mais de contraindre le langage des motifs qui couvrent les données. Individuellement, nous donnons la préférence à des motifs précis plutôt qu'à des motifs couvrants. En contrepartie, nous comptons extraire une large gamme de motifs pour représenter et couvrir exhaustivement les données. La base de données enrichie étant maintenant formellement définie, il faut nous doter de relations et mesures adéquates pour explorer les données et en extraire les motifs d'intérêt.

## 5.3 Fouille de données séquentielle hiérarchique

### 5.3.1 Hiérarchie d'items

Ainsi, la base de données est constituée d'items qui résultent d'un enrichissement des données. Or cet enrichissement, issu de l'application de ressources morpho-syntaxiques et lexicales, permet d'apporter une information de généralisation des données, qui n'étaient au départ que des tokens isolés de $\Sigma_d$. Nous pouvons généraliser les items, nous formalisons ceci comme une hiérarchie sur les items, à mettre en correspondance avec l'axe *ontologique* évoqué en section 1.1. Ceci suppose que nous puissions extraire des tokens enrichis les généralisations possibles pour n'importe quel item.

Nous avons choisi de modéliser les enrichissements par préfixation (à l'aide du `'/'`) des items. Nous pouvons alors laisser aux analyses toute latitude d'être plus ou moins fine selon les tokens considérés. Dans le cas général, extraire les préfixes des items nous permet de former l'alphabet $\Sigma_p$ des motifs, qui contient les items $\Sigma_r$ (tokens enrichis, dont les marqueurs $\Sigma_m$), et toutes leurs généralisations possibles. Par exemple, de l'item `'VER/visiter/visite'` seront extraits les préfixes (ou généralisations) `'VER/visiter'` et `'VER'`. Par concaténation, nous obtenons le langage des motifs $\mathcal{L}_p = (\Sigma_p)^*$.

Ces généralisations sont formalisées comme une relation d'ordre $\geq_{hi}$ sur $\Sigma_p$ qui nous donne la *hiérarchie* sur les items. Or, par utilisation de préfixes, il vient que cette relation est un ordre partiel et que la hiérarchie est une forêt (ensemble d'arbres) sur les items, car cet ordre respecte les propriétés suivantes :

– **Réflexivité** : $e \geq_{hi} e$,
– **Transitivité** : $e_1 \geq_{hi} e_2$ et $e_2 \geq_{hi} e_3 \Rightarrow e_1 \geq_{hi} e_3$,
– **Anti-symétrie** : $e_1 \geq_{hi} e_2$ et $e_2 \geq_{hi} e_1 \Rightarrow e_1 = e_2$,
– **Arborescence** : $e_1 \geq_{hi} e_3$ et $e_2 \geq_{hi} e_3 \Rightarrow e_2 \geq_{hi} e_1$ ou $e_1 \geq_{hi} e_2$.

Nous obtenons finalement, à partir de l'enrichissement des données et de leurs préfixes, à la fois le langage des motifs $\mathcal{L}_p$ et une forêt d'arbres sur les items induite par la relation d'ordre $\geq_{hi}$. Les arbres sur lesquels nous nous appuierons sont illustrés par les figures 5.1 et 5.2.

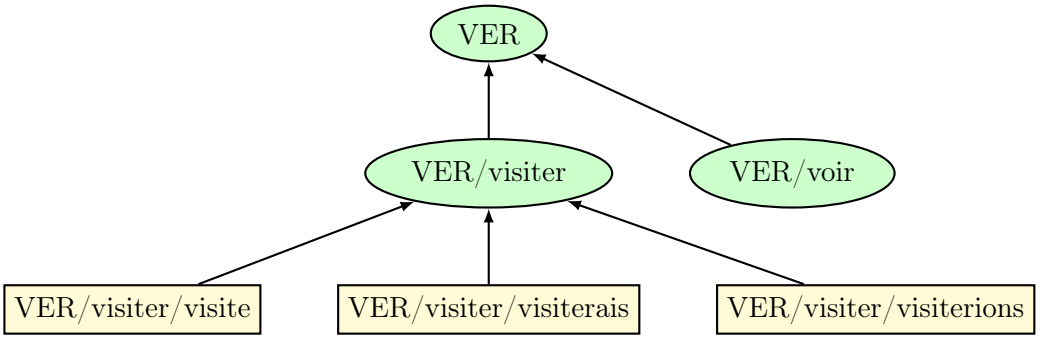



FIGURE 5.1 – Hiérarchie morpho-syntaxique pour les verbes

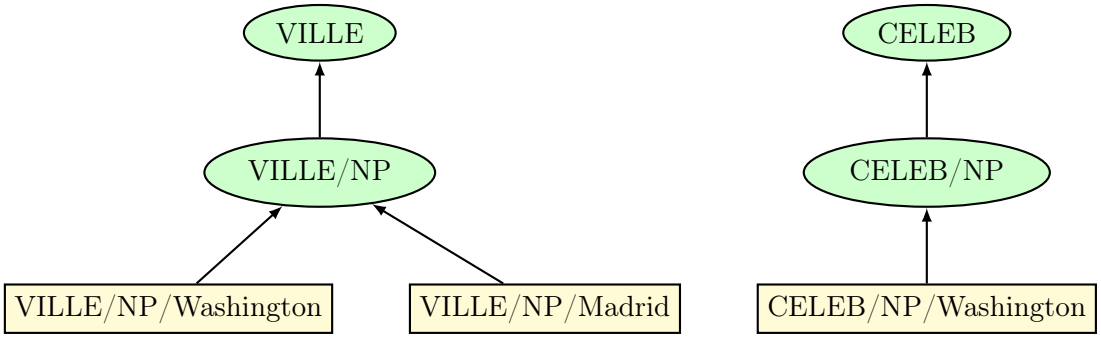



FIGURE 5.2 – Hiérarchie lexicale pour les noms propres

Ainsi, la relation d'ordre opère sur des hiérarchies locales, un ensemble d'arbres, qui sont générés suite à l'application de ressources, selon les besoins. Notons à ce sujet que, si l'utilisation de préfixes nous conduit à ce type de structure, il est possible d'un point de vue théorique, par utilisation d'un ordre partiel moins contraint, d'élargir l'approche à des ensembles de treillis, par exemple si l'on manipule des structures de traits. Encore une fois, il s'agit pour nous de limiter la combinatoire des motifs afin de mener l'exploration à son terme sur d'importants volumes de données.

### 5.3.2 Relation de couverture des motifs sur les données

Nous cherchons à explorer la base de données afin d'y découvrir des motifs qui *couvrent* les données. Dans un premier temps, ces motifs n'ayant pas immédiatement pour objectif de généraliser les données, ils sont constitués par concaténation à partir de l'alphabet $\Sigma_r$. Les motifs qui couvrent les données appartiennent donc au langage $(\Sigma_r)^*$.

Commençons par définir la manière dont un item de motif peut couvrir un item de la base de données enrichie. Il faut pour cela tenir compte de la présence des disjonctions exclusives dans les données enrichies :

**Couverture d'un item sur une donnée** : soient un item $p \in \Sigma_r$ et une donnée enrichie $i = i_1 \oplus i_2 \oplus i_3 \dots i_n$ tel que, pour tout $j$, $i_j \in \Sigma_r$, alors $p$ *couvre* $i$, noté $p \geq_{ci} i$ s'il existe au moins un $k \in [\![1, n]\!]$ tel que $i_k = p$.

Par suite, nous définissons la couverture d'un motif de la manière suivante :

**Couverture d'un motif séquentiel sur des données** : soient un motif séquentiel $P = p_1 p_2 \dots p_n \in \mathcal{L}_p$ et une séquence de la base de données enrichie de même nombre d'items $I = i_1 i_2 \dots i_n \in \mathcal{L}_r$, alors $P$ *couvre* $I$, noté $P \geq_c I$, si pour tout $j \in [\![1, n]\!]$, alors $p_j \geq_{ci} i_j$.

Par exemple, `'A C'` $\geq_c$ `'A`$\oplus$`B C'`.

Nous voyons que nous sommes en mesure d'établir une correspondance entre des motifs et des segments des données enrichies, aux ambiguïtés présentes dans les données près. Des *motifs séquentiels* (tokens simples, marqueurs, mots composés, syntagmes, etc.) peuvent dès lors être construits et confrontés aux données. Dorénavant, nous appellerons simplement *motif* un motif séquentiel, n'ayant pas à considérer d'autres types de motifs.

### 5.3.3 Relations de généralisation entre motifs

Tels quels, les motifs sont capables de *couvrir* les données enrichies. Notre objectif est maintenant de nous appuyer sur la hiérarchie pour explorer les motifs qui généralisent les données. A cet effet, nous procédons en *intention*, par examen des éléments dont ils sont composés. Commençons par comparer deux motifs de même taille, pour lesquels la hiérarchie nous permet de définir une première relation de *généralisation* :

**Généralisation hiérarchique entre motifs** : soient deux motifs $P = p_1 p_2 \dots p_n \in \mathcal{L}_p$ et $Q = q_1 q_2 \dots q_n \in \mathcal{L}_p$, alors $P$ *généralise hiérarchiquement* $Q$, noté $P \geq_h Q$, si pour tout $j \in [\![1, n]\!]$, alors $p_j \geq_{hi} q_j$.

Par exemple, `'A B C'` $\geq_h$ `'A B/D C'` $\geq_h$ `'A B/D/E C/F'`.

Ensuite, lorsque l'on considère deux motifs, de taille différentes dont l'un est sous-séquence de l'autre, nous pouvons définir une seconde relation de *généralisation* :

**Généralisation par affixation entre motifs** : soient deux motifs $P = p_1 p_2 \dots p_n \in \mathcal{L}_p$ et $Q = q_1 q_2 \dots q_p \in \mathcal{L}_p$, alors $P$ *généralise par affixation* $Q$, noté $P \geq_a Q$, si $p \geq n$ et s'il existe au moins un $k \in [\![0, p - n]\!]$ tel que, pour tout $j \in [\![1, n]\!]$, alors $q_{j+k} = p_j$.

Par exemple, `'B'` $\geq_a$ `'A B'` $\geq_a$ `'A B C'`.

La généralisation par affixation requiert que les éléments d'un motif soient contigus dans les données. Ceci nous empêche d'observer simultanément le langage naturel lui-même

(modulo les annotations) et ses corrélations avec les marqueurs d'annotation. Or nous souhaitons être en mesure de couvrir les observations à l'aide des motifs, indépendamment de la présence de *marqueurs d'annotation* supplémentaires dans les données. Par exemple, dans le texte *'Il visite <loc> Madrid </loc> .'*, nous souhaitons que, entre autres, les motifs `'visiter NP'` et `'visiter <loc> NP </loc>'` soient tous deux observés, afin d'en induire des associations entre le langage naturel et les marqueurs d'annotation.

Pour prendre cela en compte, de manière intuitive, nous recherchons *tous* les éléments d'un motif, mais considérons qu'il peut y avoir des marqueurs d'annotation *mélangés à ces éléments* dans les données. Ceci est réalisé par la définition d'une troisième relation de généralisation :

**Généralisation sur marqueurs entre motifs** : soient deux motifs $P = p_1 p_2 \dots p_n \in \mathcal{L}_p$ et $Q = q_1 q_2 \dots q_p \in \mathcal{L}_p$, alors $P$ *généralise sur marqueurs* $Q$, noté $P \geq_m Q$, si $p \geq n$ et s'il existe une fonction *discrète strictement croissante* $C()$ définie de $[\![1, n]\!]$ vers $[\![1, p]\!]$ telle que, pour tout $j \in [\![1, n]\!]$, alors $p_j = q_{C(j)}$ et, pour tout $k \in [\![1, p]\!]$ tel que $k \notin \{C(j), j \in [\![1, n]\!]\}$, alors $q_k \in \Sigma_m$.

Par exemple, `'A B C'` $\geq_m$ `'A <loc> B C'` $\geq_m$ `'<pers> A </pers> <loc> B </loc> C'`.

Nous aurons besoin, pour mettre en œuvre cette couverture, d'implémenter (avec un coût algorithmique raisonnable) l'application $C()$ qui réalise la correspondance entre les indices des motifs et ceux des données *aux marqueurs près*. Nous aborderons cet aspect du problème en section 8.1, mais remarquons dès à présent que cet appariement n'est pas nécessairement unique pour deux motifs donnés. Ceci sera résolu par recherche de l'appariement *le plus à gauche*, ou, de manière équivalente, celui qui minimisera $\sum_i C(i)$ (ce qui suffit à établir la relation de généralisation).

Les trois relations que nous avons définies sont réflexives, transitives et antisymétriques, elles forment des ordres partiels sur l'ensemble des motifs $\mathcal{L}_p$. Finalement, nous utilisons ces relations en conjonction afin de définir la relation de généralisation sur les motifs :

**Généralisation entre motifs** : soient deux motifs $P \in \mathcal{L}_p$ et $Q \in \mathcal{L}_p$, alors $P$ *généralise* $Q$, noté $P \geq_g Q$, s'il existe $R \in \mathcal{L}_p$ et $S \in \mathcal{L}_p$ tels que $P \geq_a R$, $R \geq_m S$ et $S \geq_h Q$.

Par exemple, `'<pers> A/D </pers> B'` $\geq_g$ `'<pers> A/D/E </pers> <loc> B </loc> C/F'`.

Nous obtenons ainsi, par construction, un ordre partiel sur les motifs. Nous sommes en mesure d'établir des relations de généralisation entre motifs par utilisation de hiérarchies (*ontologiquement*) et d'affixes ou de marqueurs (*structurellement*). L'ensemble des motifs est organisé sous forme de treillis. Nous nous appuierons en particulier sur la propriété de *généralisation sur marqueurs* pour déterminer quels motifs sont corrélés à la présence de *marqueurs*, interprétés comme des *instructions*. Par leurs contraintes fortes en matière de séquentialité (exigence de contiguïté, aux marqueurs près), ils explorent très localement la corrélation entre les données et les *instructions* de structuration en entités nommées.

## 5.4 Proposition en exploration de données : les segments

Nous remarquons que les données enrichies, initialement segmentées sous forme de tokens, sont enrichies selon des mécanismes (morphologie, morpho-syntaxe, lexiques) qui

peuvent opérer sur plusieurs tokens (expressions composées, voire constituants syntaxiques). Nous pourrions donc être amenés à extraire des motifs au sein desquels se trouvent des enrichissement distribués sur des *segments* de tokens. Considérons par exemple les motifs extraits sur l'énoncé *'<pers> Valéry Giscard d'Estaing </pers> succéda à [...]'*. Après enrichissements (lemmatisation, morpho-syntaxe et lexique des célébrités `'CELEB'`), nous serions en mesure d'extraire les motifs :

- `'<pers> CELEB/NP CELEB/NP CELEB/PREP/de CELEB/NP </pers> VER/succéder PREP/à'`,
- `'<pers> CELEB CELEB/NP CELEB/PREP/de CELEB/NP </pers> VER/succéder PREP/à'`,
- `'<pers> CELEB/NP CELEB CELEB/PREP/de CELEB/NP </pers> VER/succéder PREP/à'`,
- `'<pers> CELEB CELEB CELEB/PREP/de CELEB/NP </pers> VER/succéder PREP/à'`,
- ...
- `'<pers> CELEB CELEB CELEB CELEB </pers> VER/succéder PREP/à'`,
- ...

Nous voyons que l'enrichissement des données conduit à produire des motifs généralisés sur chaque token, sans tenir compte de la portée de ces enrichissements. Or, dans notre exemple *'Valéry Giscard d'Estaing'* a été catégorisé comme une entrée du lexique. De ce point de vue, nous pouvons affirmer qu'il y a là *un seul* item `'CELEB'`, suivi du verbe `'VER/succéder'`. Ou, si nous souhaitons décomposer `'CELEB'`, il nous semble qu'il y aurait un item `'CELEB/NP'`, suivi d'une préposition `'CELEB/PREP'` puis d'un autre `'CELEB/NP'` (ce qui caractériserait un nom à particule), ce bloc étant suivi du verbe `'VER/succéder'`. La généralisation de motifs fonctionnant par *segments* d'items nous donnerait :

- `'<pers> CELEB/NP CELEB/PREP/de CELEB/NP </pers> VER/succéder PREP/à'`,
- `'<pers> CELEB </pers> VER/succéder PREP/à'`,
- ...

Nous cherchons donc à éviter que les motifs ne contiennent des items contigus qui se généralisent entre eux. Dans le cas général, nous émettons l'hypothèse suivante : les motifs qui contiennent des items identiques contigus peuvent être explorés sous forme de *segments*, à la manière dont les grammaires (et les transducteurs) traitent et *réduisent* des séquence de tokens en constituants, en syntagmes, propositions. Notre hypothèse est que ceci pourrait être avantageusement pris en compte en cours de fouille de données. En corollaire, nous sommes conduits à contraindre les motifs : un marqueur ne pourra se trouver à l'intérieur d'un segment. Nous pouvons alors définir les *motifs de segments* de la manière suivante :

**Motif de segments** : un motif de segments est un motif $P = p_1 p_2 \dots p_n \in \mathcal{L}_p$ tel que, pour tout $j \in [\![1, n-1]\!]$ et $k \in [\![j+1, n]\!]$ tels que tout $l \in [\![j+1, k-1]\!]$ vérifie $p_l \in \Sigma_m$, alors $p_j \not\geq_h p_k$ et $p_k \not\geq_h p_j$.

Par exemple, `'A/C <pers> A/D B'` ou `'A/C B A'` sont des motifs de segments, tandis que `'A B B C'` ou `'A/B A'` n'en sont pas.

Cette définition stipule que, pour ces motifs, deux items contigus (ou séparés par des marqueurs) ne peuvent être identiques ou parents l'un de l'autre. Un élément d'un tel motif peut couvrir un *segment*, soit plusieurs items qu'il généralise tous. Ces motifs étant plus contraints, nous raisonnons sur un sous-langage, que nous notons $\mathcal{L}_{p^+}$ (le '+' faisant référence au symbole de répétition des expressions régulières).

Cette formulation permet donc de décrire les données par *segments*, qui pourront se situer à divers niveaux de la hiérarchie. Par exemple, si nous disposons des données `'A/C/E`

`A/C/F A/D B'`, nous pourrons les généraliser graduellement en *motifs de segments* `'A/C A/D B'` et `'A B'`. En conséquence, nous reformulons la couverture pour ces motifs (qui utilise également la couverture par item $\geq_{ci}$) de la manière suivante :

**Couverture d'un motif de segments sur des données** : soient un motif de segments $P = p_1 p_2 \dots p_n \in \mathcal{L}_{p^+}$ et une séquence de la base de données enrichie $I = i_1 i_2 \dots i_p \in \mathcal{L}_r$, alors $P$ *couvre les segments de* $I$, noté $P \geq_{c^+} I$, s'il existe une fonction *discrète croissante* $S()$ définie de $[\![1, p]\!]$ vers $[\![1, n]\!]$ telle que, pour tout $j \in [\![1, p]\!]$, alors $p_j \geq_{ci} i_{S(j)}$.

Par exemple, `'A B E'` $\geq_{c^+}$ `'A B⊕C B⊕D E'`.

L'utilisation d'une application $S()$ reflète, comme pour la généralisation sur marqueurs, la nécessité de réaliser un appariement entre les données et le motif considéré, dont l'algorithme sera donné en section 8.1. De la même manière, nous adaptons la généralisation hiérarchique pour ces motifs, afin qu'ils tiennent compte de la généralisation d'items en un *segment*. Effectivement, lorsque nous souhaitons généraliser un motif `'A B/D B/E C'`, nous obtenons le motif `'A B C'` :

**Généralisation hiérarchique entre motifs de segments** : soient deux motifs de segments $P = p_1 p_2 \dots p_n \in \mathcal{L}_{p^+}$ et $Q = q_1 q_2 \dots q_p \in \mathcal{L}_{p^+}$, alors $P$ *généralise hiérarchiquement les segments de* $Q$, noté $P \geq_{h^+} Q$, s'il existe une une fonction *discrète croissante* $S()$ définie de $[\![1, p]\!]$ vers $[\![1, n]\!]$ telle que, pour tout $j \in [\![1, p]\!]$, alors $p_j \geq_{ci} q_{S(j)}$.

Par exemple, `'A B C'` $\geq_{h^+}$ `'A B C/G C/H'` $\geq_{h^+}$ `'A B/D B/E B/F C/G C/H'`.

Les deux autres relations de généralisation (par affixation, sur marqueurs) restent inchangées. Tant que nous nous restreignons au langage des *motifs de segments* $\mathcal{L}_{p^+}$, l'ordre partiel est préservé. Et nous pouvons simplement reformuler la généralisation pour les motifs de segments en :

**Généralisation entre motifs de segments** : soient deux motifs $P \in \mathcal{L}_{p^+}$ et $Q \in \mathcal{L}_{p^+}$, alors $P$ *généralise les segments de* $Q$, noté $P \geq_{g^+} Q$, s'il existe $R \in \mathcal{L}_{p^+}$ et $S \in \mathcal{L}_{p^+}$ tels que $P \geq_a R$, $R \geq_m S$ et $S \geq_{h^+} Q$.

Par exemple, `'A <pers> B <pers>'` $\geq_{g^+}$ `'A <pers> B/D B/E <pers> <loc> C <loc>'`.

Cette formalisation alternative des motifs sera expérimentée en comparaison avec les motifs séquentiels standards exposés plus haut. L'objectif reste d'explorer les données, comme précédemment, en s'appuyant sur l'*ontologie* (hiérarchie) et la *structure* (affixe, marqueurs) simultanément. En travaillant sur des *segments*, nous cherchons à tisser un lien entre les approches *orientées données*, qui fonctionnent généralement par tokens, et les approchées *orientées connaissances* (en particulier les transducteurs), qui construisent des représentations incrémentales par réduction, à l'aide de grammaires.

## 5.5 Les motifs d'intérêt pour l'annotation par marqueurs

Nous disposons maintenant d'un formalisme qui nous indique comment les motifs couvrent les données depuis $\mathcal{L}_d$ vers $\mathcal{L}_p$ et comment ils se généralisent au sein de $\mathcal{L}_p$. A partir de corpus volumineux, nous allons pouvoir explorer des données en faisant abstraction pour partie de la variabilité naturelle du langage. Notre objectif est d'obtenir des motifs fiables (précis pour la reconnaissance d'entités nommées) et robustes (généralisés, tolérants à la présence de bruit) pour la reconnaissance d'entités nommées [Nouvel et Sou-

let, 2011]. Cependant, la combinatoire rend trop fastidieuse l'énumération exhaustive de tous les motifs possibles pour tous les énoncés qui nous serons donnés en exemple. Il nous faut des critères permettant de guider la recherche de motifs, des mesures qui quantifient *l'intérêt* qu'ils présentent a priori.

De nombreux travaux ont cherché à établir empiriquement quelles mesures sont les plus adéquates pour extraire des connaissances [Tan *et al.*, 2002]. L'objectif est généralement d'orienter l'exploration des données vers les représentations (motifs, règles) qui maximisent ces mesures. Pour notre problématique, nous formulons des contraintes en faisant reposer nos critères de sélection sur quatre éléments :

- **Présence de marqueurs** : pour réaliser la reconnaissance d'entités nommées, les motifs devront contenir des *marqueurs*,
- **Fréquence minimale** : afin de ne pas relever de cas particuliers, les motifs devront *couvrir* un nombre minimal d'occurrences dans les corpus d'extraction,
- **Confiance minimale** : pour être *productifs* en reconnaissance d'entités nommées, les motifs devront révéler une corrélation entre le langage naturel (données enrichies, hors marqueurs) et la présence de marqueurs,
- **Non-redondance** : afin de limiter le nombre de motifs extraits, nous sélectionnons les éléments les plus *informatifs* au sein des ensembles de motifs couvrant les mêmes exemples.

### 5.5.1 Règles d'annotation

L'exploration des données enrichies va nous permettre de recenser, potentiellement, un grand nombre de motifs. Parmi ceux-ci, certains contiendront les *marqueurs d'annotation* qui nous intéressent. Dans la lignée de notre proposition concernant l'approche que nous proposons (c.f. 3.3), nous pouvons ici faire un lien avec les *transducteurs*, qui examinent des textes afin d'insérer des balises en leur sein. Encore une fois, si nous nous appuyons sur les données, nous ne formulons pas notre approche comme l'attribution d'une classe aux items d'une séquence.

Le cadre théorique de la fouille de données nous permet de modéliser des *règles d'association* qui recherchent des relations d'implications entre motifs, en se basant sur les observations [Budi et Bressan, 2007]. Par exemple, au regard de la donnée enrichie `'PRO/il/Il VERB/visiter/visite <loc> VILLE/NP/Madrid </loc>'`, nous souhaiterions, entre autres, considérer l'implication suivante :

`'VERB/visiter VILLE/NP'` $\Rightarrow$ `'VERB/visiter <loc> VILLE/NP </loc>'`

En d'autres termes, la présence du verbe `'VERB/visiter'` suivi d'un token enrichi à l'aide des lexiques en `'VILLE/NP'` conduit à l'ajout, par un mécanisme similaire à la transduction, de deux *marqueurs d'annotation* autour de ce dernier item. Remarquons cependant que, dans notre formalisation, les données enrichies ne distinguent que l'alphabet $\Sigma_m$ des *marqueurs d'annotation*, sans contraintes sur le bon agencement de ces éléments. De notre point de vue, cela permet d'établir un lien entre le langage et les *instructions* locales, sans que les motifs soient nécessairement contraints à les explorer par paires. Par exemple, au vu des énoncés *'Il rencontra Georges Pompidou'* et *'Nous rencontrons Valéry Giscard d'Estaing'*, nous serons amenés à considérer l'association :

`'VERB/rencontrer CELEB/NP'` ⇒ `'VERB/rencontrer <pers> CELEB/NP'`

Ainsi, le verbe `'VERB/rencontrer'` suivi d'un item `'CELEB/NP'` peut être lié à la borne gauche d'une entité de type personne. Ce type de motif, loin d'être improductif, peut participer à la construction d'une structuration en entité nommée. Nous chercherons à en tirer parti et à démontrer expérimentalement que ces motifs *partiels* sont plus efficaces que des motifs contraints de reconnaître les deux balises des entités (alors dits *complets*). Notre hypothèse est que les motifs partiels permettront de décomposer la reconnaissance d'entités nommées en recherche séparées des *marqueurs* individuels (les *instructions* locales). Comment utiliser automatiquement ces motifs pour former une annotation valide sera considéré ultérieurement.

En conséquence, nous définissons une règle d'annotation par la partie droite de ces implications (la partie gauche pouvant en être déduite directement par retrait des marqueurs) :

**Règle d'annotation** : une règle d'annotation $R$ est un motif $P = p_1 p_2 \dots p_n \in \mathcal{L}_p$ tel qu'il existe au moins un $j$ pour lequel $p_j \in \Sigma_m$ et un $k$ pour lequel $p_k \notin \Sigma_m$.

L'exploration des données est alors bien peu contrainte pour rechercher ces règles : il suffit qu'un motif contienne un marqueur et un item du langage naturel pour qu'il puisse potentiellement être sélectionné comme *règle d'annotation*. L'inconvénient est qu'il y aura de très nombreux motifs à considérer lors de la fouille. Mais l'avantage sera que nous explorerons les données de manière très objective (selon les enrichissements mis en œuvre) et que d'autre contraintes pourront aisément être adjointes à celle-ci. Nous nous focalisons sur l'extraction de ces *règles d'annotation*, sans pour autant que cela nous épargne d'explorer plus largement les données, notamment pour calculer la confiance associée à ces règles.

### 5.5.2 Fréquence minimale

La relation de généralisation a pour objectif de permettre l'exploration de motifs qui ne sont pas trop spécifiques aux données explorées, mais qui, par leur nombre, captent néanmoins la diversité d'expressions linguistiques liées aux marqueurs d'annotation. Pour ce faire, nous considérons, a priori, qu'une règle ne doit pas être un cas particulier du corpus exploré. Il nous faut alors être en mesure de comptabiliser le nombre d'occurrences concernées par un motif donné. A cet effet, nous utilisons la relation de couverture des motifs sur les données, afin de déterminer l'ensemble des *occurrences* d'un motif pour un énoncé :

**Occurrences d'un motif pour un énoncé** : soient un motif $P = p_1 p_2 \dots p_n \in \mathcal{L}_p$ et un énoncé de la base de données enrichie $I = i_1 i_2 \dots i_p \in \mathcal{L}_r$, alors les occurrences de $P$ dans $I$, noté $Occ(P, I)$, correspondent aux paires d'indices $(j, k)$ tels qu'il existe un motif $Q$ pour lequel $Q \geq_c i_j \dots i_k$ et $P \geq_g Q$ et qu'il n'existe pas de motif $R \neq Q$ et de couple d'indices $(j', k')$ pour lesquels $j' \geq j$, $k' \leq k$, $R \geq_g Q$, $R \geq_c i_{j'} \dots i_{k'}$ et $P \geq_g R$.

La définition ci-dessous donne la liste des indices correspondants aux occurrences d'un motif au sein d'un énoncé. Pour ce faire, nous déterminons, pour un motif $P$ et une donnée $I$, un motif intermédiaire $Q$ qui, simultanément, *couvre les données* de $I$ et *se généralise* en $P$. Nous assortissons ceci d'une contrainte qui stipule que $Q$ ne doit pas pouvoir se

généraliser en un autre motif $R$ qui couvrirait une portion de ces mêmes données et se généraliserait également en $P$. Ceci nous garantit que, tout en tenant compte de la généralisation, un motif ne soit cependant associé qu'à ses occurrences minimales. Si, pour illustration, nous disposons de la données $I =$ `'A B A'` , alors le motif $P =$ `'A'` aura pour occurrences $Occ(P, I) = \{(1, 1), (3, 3)\}$, l'occurrence $(1, 3)$ (par affixation) n'étant pas minimale.

Finalement, nous pouvons déterminer la fréquence absolue d'un motif dans la base de données $\mathcal{D}$ comme la somme des cardinalités de ces ensembles d'occurrences pour chaque énoncé de la base enrichi à l'aide de la fonction d'enrichissement $R()$ :

**Fréquence absolue d'un motif dans une base de données** : soit un motif $P$ et une base de données $\mathcal{D}$, alors la fréquence absolue de $P$ dans $\mathcal{D}$ est donnée par :

$$Freq_{abs}(P, \mathcal{D}) = \sum_{E \in \mathcal{D}} |Occ(P, R(E))|$$

Par cette formule, nous nous munissons d'une mesure qui nous indique la représentativité des règles extraites. Un seuil pourra alors être défini afin de n'explorer, parmi les motifs, que ceux qui ont une fréquence absolue minimale dans une base de données. Nous chercherons cependant à déterminer un seuil adéquat indépendamment de la taille de la base de données (nombre d'énoncés et longueur des énoncés), ainsi nous utiliserons également la *fréquence relative*, notée par simplification $Freq$, qui tient compte de la taille en nombre d'items des énoncés (notée $|E|$ pour un énoncé $E$) :

**Fréquence relative d'un motif dans une base de données** : soit un motif $P$ et une base de données $\mathcal{D}$, alors la fréquence relative de $P$ dans $\mathcal{D}$ est donnée par :

$$Freq(P, \mathcal{D}) = \frac{Freq_{abs}(P, \mathcal{D})}{\sum\limits_{E \in \mathcal{D}} |E|}$$

Cette mesure prend ses valeurs dans $[0, 1]$ (la fréquence d'un motif ne pouvant dépasser le nombre d'items des énoncés dans la base). Nous pouvons également l'interpréter comme liée à la probabilité d'apparition d'un motif à une position quelconque d'un énoncé. En corollaire, nous vérifions que, par construction, la fréquence est *anti-monotone* : pour deux motifs $P$ et $Q$ tels que $P \geq_g Q$ et pour un énoncé de la base de donnés enrichie $I$ alors nous montrons que $|Occ(P, I)| \geq |Occ(Q, I)|$. Effectivement, pour n'importe quelle occurrence $(j, k) \in Occ(Q, I)$, il existe alors un motif $R$ tel que $R \geq_c i_j \dots i_k$ et $Q \geq_g R$. Nous considérons alors les cas particuliers de la relation de généralisation :

- si $P \geq_h Q$ alors, par transitivité de $\geq_g$, $(j, k) \in Occ(P, I)$,
- si $P \geq_a Q$ alors il existe $R'$ et $(j', k')$ tels que $j' \geq j$, $k' \leq k$ et $(j', k') \in Occ(P, I)$,
- si $P \geq_m Q$ et $Q \not\geq_a P$ alors, par transitivité de $\geq_g$, $(j, k) \in Occ(P, I)$.

Ainsi, dans le premier et le troisième cas (lorsque la généralisation sur marqueurs ne concerne pas de marqueurs en début ou fin de motif), nous voyons qu'une occurrence de $Q$ sera également occurrence de $P$. Pour la généralisation par affixation, il existera au moins une occurrence au sein de $i_j \dots i_k$ couverte par $P$. Comme notre relation $\geq_g$ est combinaison de ces trois relations de généralisation, nous en déduisons de manière générale que le nombre d'occurrences d'un motif croît lorsqu'il est généralisé. Et qu'inversement, un motif

ne saurait être plus fréquent qu'une de ses généralisations. Ceci nous sera particulièrement utile pour implémenter l'exploration des motifs fréquents que nous décrirons en section 8.2.

### 5.5.3 Confiance minimale

Même lorsqu'une règle est fréquente, ceci ne signifie pas pour autant qu'elle soit pertinente au regard des marqueurs d'entités nommées. En effet, il est plausible qu'un motif tel que `'VERB <loc>'`, signifiant qu'un marqueur de début de lieu peut suivre un verbe, dispose de suffisamment d'occurrences dans un corpus pour être extrait. Intuitivement, nous savons que la présence d'un verbe est un indice bien faible pour reconnaître une entité nommée de type lieu. Au delà de la *description* de motifs qui contiennent les marqueurs, nous souhaitons observer ceux qui permettent d'en faire la *prédiction*. Il nous faut donc mesurer, pour une règle, la corrélation entre les parties du motif qui relèvent du langage naturel et leurs co-occurrences avec les de marqueurs. A cet effet, rappelons que la relation de *généralisation sur marqueurs*, $\geq_m$, nous permet de relever les occurrences d'un motif *aux marqueurs près* : par exemple, parmi les occurrences du motif `'NP VERB'` sont également comptabilisées celles du motif `'<pers> NP </pers> VERB'`.

En premier lieu, nous définissons une fonction qui retourne, pour une règle, le motif correspondant lorsque les marqueurs sont omis, à l'aide de la *généralisation sur marqueurs* :

**Retrait des marqueurs d'une règle d'annotation** : soit une règle d'annotation $P$, alors la fonction $Ret_m(P)$ renvoie le motif $Q \in (\Sigma_p/\Sigma_m)^*$ tel que $Q \geq_m P$.

Comme $\Sigma_p$ est le langage enrichi et $\Sigma_m$ celui des marqueurs, alors $(\Sigma_p/\Sigma_m)^*$ est le langage des motifs qui ne contiennent pas de marqueurs. Il vient immédiatement que, pour une règle $P$, il n'existe qu'un seul motif qui corresponde à cette règle lorsque les marqueurs sont omis $Q = Ret_m(P)$. A l'aide de cette opération, nous sommes en mesure de déterminer, pour une règle d'annotation, son nombre d'occurrences lorsque les marqueurs sont omis. En le rapportant au nombre d'occurrences de la règle (marqueurs compris), nous obtenons une mesure de confiance, à la manière dont cette mesure est calculée pour les règles d'association [Min, 1993, Budi et Bressan, 2007]. Nous formulons donc cette métrique de la manière suivante :

**Confiance d'une règle d'annotation dans une base de données** : soit une règle d'annotation $P$ et une base de données $\mathcal{D}$, alors la confiance de $P$ dans $\mathcal{D}$ est donnée par :

$$Conf(P, \mathcal{D}) = \frac{Freq(P, \mathcal{D})}{Freq(Ret_m(P), \mathcal{D})}$$

Notons que comme, par définition, $P \geq_m Ret_m(P)$, alors $Freq(P, \mathcal{D}) \leq Freq(Ret_m(P), \mathcal{D})$ et la confiance prend ses valeurs dans l'intervalle $[0, 1]$. Nous obtenons ainsi une mesure normalisée qui évalue le degré de corrélation entre une règle d'annotation et les marqueurs qu'elle introduit. Moins formellement, cette statistique peut être interprétée comme la probabilité qu'une règle d'annotation prédise correctement ses marqueurs d'annotation. Ainsi, nous pouvons sélectionner les règles selon leur confiance, ce qui correspond empiriquement à leur capacité à construire une annotation en entités nommées correcte.

### 5.5.4 Règles informatives

Sélectionner les motifs afin de ne retenir que les *règles d'annotation* qui ont une *fréquence* et une *confiance* minimales ne nous épargne pas la combinatoire liées aux données et aux enrichissements. Effectivement, lorsque nous fouillons les données, le nombre de motifs qu'il faudra explorer est lié simultanément au nombre de motifs qui *couvrent* les données et au nombre de *généralisations* qu'ils produisent. Ce nombre de motifs à prendre en considération peut devenir rapidement très conséquent. Mais, intuitivement, nous pressentons que ces motifs peuvent être très redondants, en particulier lorsqu'ils couvrent les mêmes occurrences dans les données.

Prenons en exemple les données $I_1$ = `'A/E B/F C/G'` , $I_2$ = `'A/E B/F C/H'` et $I_3$ = `'A/E B/F D/G'` dans lesquelles nous cherchons les motifs de taille $n$ qui ont exactement $f$ occurrences (ici en fréquence absolue). Les motifs considérés sont présentés en table 5.1, dont ceux qui couvrent directement (sans généraliser) les données sont distingués par une astérisque.

| | $n = 1$ | $n = 2$ | $n = 3$ |
|---|---|---|---|
| $f = 2$ | `'C'` | `'B C'`<br>`'B/F C'` | `'A B C'`<br>`'A/E B C'`<br>`'A B/F C'`<br>`'A/E B/F C'` |
| $f = 3$ | `'A'`<br>`'A/E'`*<br>`'B'`<br>`'B/F'`* | `'A B'`<br>`'A/E B'`<br>`'A B/F'`<br>`'A/E B/F'`* | |

Table 5.1 – Redondance de motifs extraits

Sur cet exemple minimal, nous voyons que peu de motifs couvrent directement les données. Pourtant, potentiellement, cinq items couvrent directement les données (`'A/E'`, `'B/F'`, `'C/G'`, `'C/H'` et `'D/G'`) et la combinatoire conduirait à explorer, à taille $n$, $5^n$ motifs possibles. Nous constatons que pour $n = 3$ et $f \geq 2$, seuls 4 motifs sont fréquents, au lieu de 125. Ceci nous confirme qu'imposer un seuil de fréquence minimale est efficace pour guider l'exploration. Ce n'est cependant pas suffisant, car nous remarquons qu'avec les généralisations, la combinatoire joue à plein au sein de $(\Sigma_p)^*$ selon la taille des motifs. Dans l'exemple, pour $n = 2$ et $f = 3$ le motif `'A/E B/F'` se généralise en `'A B'`, `'A/E B'`, `'A B/F'`) qui concernent tous les mêmes occurrences dans les données.

Nous pouvons illustrer cet effet avec des motifs dédiés à la reconnaissance des entités nommées. Par exemple, il paraît plausible qu'un motif tel que `'<pers> CELEB/NP </pers> VERB/avoir VERB/rencontrer <pers> CELEB/NP </pers> PREP/à <loc> VILLE/NP </loc>'` (pour plus de lisibilité, nous y omettons les tokens) soit suffisamment fréquent pour être extrait. Or, si l'on considère uniquement les généralisations de ses items par la hiérarchie, il génère $2^6 - 1 = 63$ généralisations. Et ce chiffre sera démultiplié lorsque l'on tiendra compte des généralisations par affixe ou sur marqueurs. Et parmi ces généralisations, celles qui ont les mêmes occurrences dans les données sont empiriquement redondantes.

En somme, lors de l'extraction des règles à partir des données, il n'y a pas de difficulté

majeure à relever les motifs, même relativement longs, qui *couvrent* directement les données pour un seuil de fréquence fixé. Remarquons à ce sujet que la problématique n'est pas spécifique à la fouille de données : les approches orientées données présentés en section 3.2, en particulier les CRF, utilisent couramment des *fenêtres* sur les observations (qui peuvent parfois paraître arbitraires) afin d'être en mesure de paramétrer le modèle dans des temps raisonnables. Dans notre approche, nous cherchons plutôt à comparer et filtrer les motifs parmi les combinaisons engendrées par *généralisation*.

Pour ce faire, nous pouvons exprimer la contrainte d'anti-monotonie en extension, par comparaison des occurrences de motifs dans les données, que nous formulons à l'aide du théorème suivant :

**Occurrence et généralisation pour un énoncé** : soient deux motifs $P \in \mathcal{L}_p$ et $Q \in \mathcal{L}_p$ tels que $P \geq_g Q$ et un énoncé de la base de données enrichie $I = i_1 i_2 \dots i_n \in \mathcal{L}_r$, alors pour tout $(j,k) \in Occ(Q,I)$, il existe au moins un $(j',k')$ tel que $(j',k') \in Occ(P,I)$, $j' \geq j$ et $k' \leq k$

Ceci nous permet de raisonner sur les occurrences des motifs lorsqu'ils entretiennent une relation généralisation. Nous dirons ainsi que deux motifs $P$ et $Q$, tels que $P \geq_g Q$, qui ont même fréquence, ont nécessairement les mêmes occurrences dans les données *aux indices près*. Par abus de langage, ils *couvrent* les mêmes exemples et présentent une forme de *redondance*. Plus formellement, nous nous appuyons sur le théorème précédent pour définir une relation d'équivalence entre motifs selon leurs fréquences au sein d'une base de données de la manière suivante :

**Équivalence de motifs au regard d'une base de données** : soient $P$ et $Q$ deux motifs et $\mathcal{D}$ une base de données, alors $P$ est *équivalent à $Q$ au regard de $\mathcal{D}$*, notée $P \equiv_{\mathcal{D}} Q$, si $P \geq_g Q$ ou $Q \geq_g P$ et $Freq(P,\mathcal{D}) = Freq(Q,\mathcal{D})$

Cette relation d'équivalence, peut facilement être déclinée sous une forme moins contraignante en considérant les motifs qui couvrent *quasiment*, les mêmes occurrences *à quelques occurrences près*. Ainsi, deux motifs qui couvrent, par exemple à 75%, les mêmes exemples, pourront être considérés équivalent à 25% près. Nous définissons alors de consort :

**Équivalence de motifs au regard d'une base de données à $\delta$% près** : soient $P$ et $Q$ deux motifs et $\mathcal{D}$ une base de données, alors $P$ est *équivalent à $Q$ au regard de $\mathcal{D}$* à $\delta$% près, notée $P \equiv_{\mathcal{D},\delta} Q$, si $P \geq_g Q$ ou $Q \geq_g P$ et $\dfrac{|Freq(P,\mathcal{D}) - Freq(Q,\mathcal{D})|}{max(Freq(P,\mathcal{D}), Freq(Q,\mathcal{D}))} \leq \delta$

Nous remarquons que cette relation d'équivalence est, par définition de $\geq_g$, réflexive, symétrique et transitive. Ceci va nous permettre de grouper les motifs qui portent sur les mêmes occurrences *aux indices près*. Et ainsi, nous pouvons filtrer les motifs en ne choisissant, pour chaque classe, que le motif qui la *représentera* le mieux. Ce choix est évidemment dépendant de la tâche envisagée et des propriétés (précision, robustesse) que l'on souhaite donner aux motifs. Nous définissons pour chaque classe les motif les plus spécifiques, dits *maximaux*, ainsi que les plus génériques, dits minimaux :

**Motif maximal au regard d'une base de données** : soient $P$ un motif et $\mathcal{D}$ une base de données, alors $P$ est *maximal au regard de $\mathcal{D}$* s'il n'existe aucun motif $Q$ tel que $P \geq_g Q$ et $P \equiv_{\mathcal{D}} Q$

**Motif minimal au regard d'une base de données** : soient $P$ un motif et $\mathcal{D}$ une base de données, alors $P$ est *minimal au regard de $\mathcal{D}$* s'il n'existe aucun motif $Q$ tel que

$Q \geq_g P$ et $P \equiv_{\mathcal{D}} Q$

En fouille de données, les motifs maximaux sont également appelés *fermés* ou *clos*, les minimaux, *libres*. Dans l'exemple précédent, le motif `'A/E B/F'` est *maximal* pour $f = 3$ ainsi que `'A/E B/F C'` pour $f = 2$, tandis que `'A'` et `'B'` sont *minimaux* pour $f = 3$ de même que `'C'` pour $f = 2$. Selon la structure du treillis des motifs, il peut y avoir de nombreux *minimaux* ou *maximaux*. Dans le cadre théorique que nous avons choisi, comme la hiérarchie sur les items est une forêt, il ne peut y avoir qu'un seul *maximal* par classe d'équivalence. Effectivement, d'après les occurrences concernées par une classe d'équivalence, pour toutes combinaisons de $\geq_h$, $\geq_a$ et $\geq_m$, s'il existe deux relations $P \geq_g Q$ et $P \geq_g R$, alors il existe nécessairement un $S$ tel que $Q \geq_g S$, $R \geq_g S$ et bien sûr $P \geq_g S$.

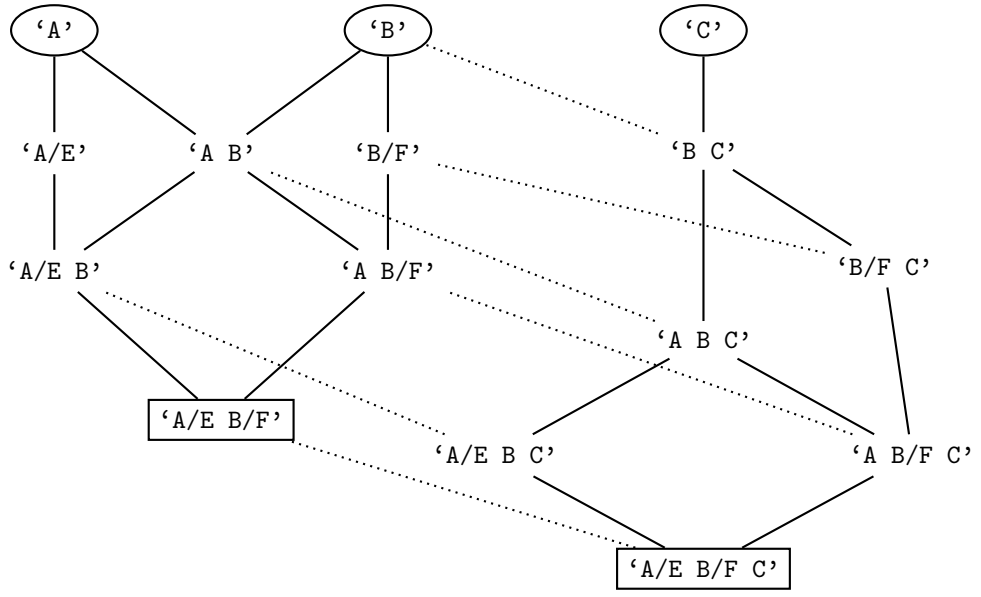



FIGURE 5.3 – Classes d'équivalence des motifs

La figure 5.3 illustre ces classes d'équivalence pour notre exemple sous la forme d'un diagramme de Hasse. Nous y voyons les relations de généralisation (arcs), dont certaines sont des relations d'équivalence (arcs pleins). Les minimaux y sont représentés par des ellipses, les maximaux par des rectangles. C'est ainsi que sera partitionné notre espace de motifs, il reste donc à sélectionner le motif le plus pertinent pour chaque classe d'équivalence. Notons également que les classes d'équivalence *partitionnent les motifs selon leurs fréquences* : cela n'interdit pas pour autant de sélectionner à la fois un motif et une de ses généralisations (ou spécialisations), du moment que leurs fréquences différent.

Dans notre cas, nous nous intéressons plus particulièrement aux motifs qui nous apportent le plus d'*information* pour produire des annotations. A cet effet, nous choisissons le motif le plus spécifique (le moins général) parmi toutes les alternatives équivalentes, ce qui correspond au motif maximal de chaque classe d'équivalence. Et en conséquence, nous sélectionnons ainsi les motifs qui contiennent le plus de *marqueurs* comme règles d'annotation.

Ainsi, nous focalisons notre recherche sur les *règles d'annotation fréquentes, confiantes et maximales au regard de la base de données*, que nous appelons *règles informatives* :

**Règle informative** : soient $P$ un motif, $\mathcal{D}$ une base de données, $F \in [0,1]$ un seuil de fréquence et $C \in [0,1]$ un seuil de confiance, alors $P$ est une *règle informative* si $P$ est une règle d'annotation, si $P$ est maximal au regard de $\mathcal{D}$, si $Freq(P,\mathcal{D}) \geq F$ et si $Conf(P,\mathcal{D}) \geq C$

Nous voyons que la formalisation proposée décrit un cadre qui se veut objectif pour explorer les données. Dans la ligne de notre perspective de *structuration* à l'aide d'*instructions locales*, nous avons soigneusement évité, alors que nous nous appuyons sur les données, de considérer que la présence de *marqueurs d'annotation* ne se traduise comme une classification de items. Nous décomposons la problématique comme une exploration des données, simultanément sur l'axe *ontologique* (hiérarchie) et sur l'axe *structurel* (motifs séquentiels et contigus, présence de marqueurs) des données. Celle-ci nous conduit à extraire des *règles d'annotation* relatives aux marqueurs des annotations en entités nommées. Quoique contrainte, l'extraction des *règles informatives* nous permettra de constituer une ressource à la fois riche et précise pour mettre en œuvre la reconnaissance des entités nommées.

# Chapitre 6

# Exploiter les règles d'annotation au sein d'un modèle numérique

## 6.1 De l'utilisation des règles extraites

Comme nous l'avons mentionné en section 3.2 et 4.3, à notre connaissance peu de travaux font un lien direct entre l'exploration en profondeur des données (fouille, motifs) et l'utilisation des motifs pour ajuster automatiquement les paramètres de systèmes pour des tâches dédiées (inférence bayésienne, régression logistique, SVM, CRF). La majorité des approches tiennent effectivement pour acquis que les motifs extraits sont déterministes et peuvent, au mieux, être ordonnés [Liu *et al.*, 1998].

Comme exposé en chapitre 3, nous cherchons à faire un rapprochement entre l'exploration exhaustive de données symboliques et le paramétrage automatique de systèmes numériques. Les premiers se focalisent sur l'extraction de connaissances complexes que l'humain pourra étudier (règles). Les seconds ajustent les poids liés des fonctions caractéristiques (traits discriminants) selon une fonction d'erreur à minimiser. Dans de nombreux cas, l'articulation entre les deux est réalisée manuellement : un ensemble de règles jugées intéressantes par un expert pourront être ajoutée comme traits disponibles pour un système orienté données (par exemple, les *motifs* à fenêtres d'un CRF). Nous réalisons cette étape automatiquement.

L'exploration des données nous permet d'alimenter un système en règles qui encodent une connaissance riche. Ces règles se fondent sur la séquentialité et la contiguïté des éléments qui composent les motifs. Hormis cela, le langage dont nous nous sommes doté est relativement peu contraint. De ce fait, l'adéquation des règles à la reconnaissance d'entités nommées repose sur la pertinence des enrichissements qui forment la hiérarchie des items et des critères retenus pour sélectionner et filtrer les motifs. Une fois les règles extraites, rien ne nous empêche de les utiliser telles quelles pour réaliser la tache d'annotation visée. Reste donc à déterminer selon quel modèle les règles seront mises en application pour réaliser l'annotation.

Immédiatement, nous voyons qu'il est possible d'appliquer les règles telles qu'elles ont été extraites, c'est à dire selon la corrélation qu'elles établissent entre le langage naturel

et les marqueurs insérés dans le cadre d'une *annotation*. Pour une règle d'annotation $T$, la fonction $Ret_m(T)$ indique le motif du langage naturel qu'il faut rencontrer pour que la règle s'applique. Par exemple, $T =$ `'VERB/rencontrer <pers> CELEB/NP CELEB/NP </pers>'` peut potentiellement s'appliquer partout où sera rencontré $Ret_m(T) =$ `'VERB/rencontrerCELEB/NP CELEB/NP'` . Nous voyons alors la possibilité de réaliser, comme pour les systèmes orientés connaissances présentés en 3.1, des *transductions*, c'est à dire l'insertion de marqueurs au sein du texte afin de réaliser l'annotation en entités nommées.

A cet effet, nous définissons le mécanisme de transduction de la manière suivante :

**Transduction d'un énoncé par une règle d'annotation** : soient une séquence de la base de données enrichie $I = i_1 i_2 \dots i_n$ et une règle d'annotation $T = t_1 t_2 \dots t_p$, alors $M = M_1 \dots M_{n+1}$ est une transduction de $I$ par $T$ et s'il existe $(j, k) \in Occ(Ret_m(T), I)$ tel que $T \geq_h M_j Ret_m(i_j) M_{j+1} \dots M_k Ret_m(i_k) M_{k+1}$ et pour tout $l < j$ et $l > k + 1$, $M_l = \emptyset$.

Ainsi, nous définissons la transduction comme l'affectation de marqueurs au sein d'un énoncé. Pour cela, nous requérons que la règle d'annotation, lorsque les marqueurs sont omis, dispose d'une occurrence dans l'énoncé. De plus, nous contraignons cette affectation de telle sorte que l'alternance des marqueurs insérés et des items de l'énoncé se généralise hiérarchiquement en la règle d'annotation. La séquence obtenue par transduction sera $M_1 i_1 M_2 i_2 \dots M_n I_n M_{n+1}$, qui appartient au langage des données enrichies $\mathcal{L}_r$. Notons que chaque $M_j$ peut contenir plusieurs marqueurs à insérer pour une position au sein de l'énoncé, comme par exemple pour la règle d'annotation `'<fonc> DET/le NC/président </fonc> <pers> CELEB/NP CELEB/NP </pers>'`.

Nous nous en tenons à cette définition et n'entrons pas ici dans des discussions plus approfondies sur le principe des transductions. Mentionnons simplement que nous réalisons au plus une seule transduction par occurrence associée dans un énoncé à chaque règle d'annotation et que nous affectons, s'il y a litige, les marqueurs le plus à gauche possible, tout en tenant compte de contraintes liées aux guides d'annotation. De manière plus générale, nous savons que pour un énoncé donné, si nous disposons nombreuses règles d'annotation extraites, il y a nécessité de prendre une décision sur les transductions possibles, ce qui peut être réalisé de multiples manières.

## 6.2 Annoter par règles selon leur confiance

Nous sommes donc, à partir des règles d'annotation, en mesure de réaliser des transductions sur un énoncé. Nous pourrions naïvement chercher à réaliser toutes les transductions possibles et considérer que l'on a obtenu une annotation. Ce principe est mis en difficulté pour les deux raisons suivantes :

- **Règles partielles** : jusque là, nous n'avons pas émis de contraintes sur la formation des marqueurs au sein d'une règle d'annotation, l'application d'une règle d'annotation peut conduire à introduire un seul marqueur, des marqueurs de types différents, etc.
- **Imbrications et chevauchements** : réaliser une transduction ne tient pas compte des marqueurs déjà insérés dans l'énoncé et peut aboutir à une annotation avec des chevauchements ou des imbrications, qui sont généralement interdites par les guides d'annotation.

Nous le voyons, chacune de ces deux raisons empêche d'appliquer toutes les règles d'annotation systématiquement. La problématique est ici de produire en sortie une annotation qui soit conforme aux règles édictées dans le guide d'annotation. Il est trivial de montrer que les règles partielles ne produisent pas une annotation valide. Pour les imbrications et chevauchements, nous faisons remarquer que même l'application des règles les plus simples peut être problématique. Par exemple réaliser toutes les transductions de la règle `‘<pers> CELEB/NP CELEB/NP </pers>’` sur les items *‘CELEB/NP/Léopold CELEB/NP/Sédar CELEB/NP/Senghor’* produira l'annotation `‘<pers> CELEB/NP/Léopold <pers> CELEB/NP/Sédar </pers> CELEB/NP/Senghor </pers>’`.

Un mécanisme de contrôle est donc indispensable pour réaliser les transductions. En première approche, nous nous inspirons du mode de fonctionnement des transducteurs : appliquer les règles d'annotation selon un ordre prédéfini, sur les occurrences les plus à gauche, tant qu'elles reconnaissent des entités nommées et produisent une annotation *valide*. Pour ce faire, nous écartons les règles dites *partielles* en ne sélectionnant que celles dont l'application produit une annotation conforme au guide. Enfin, la confiance nous paraît être une mesure naturellement adéquate à utiliser pour ordonnancer les règles. Nous rapportons en chapitre 9 les résultats obtenus avec ce premier modèle modèle utilisant les règles d'annotation pour reconnaître les entités nommées.

Cependant, outre le fait d'écarter les règles *partielles*, ordonner les règles par leur confiance paraît intuitivement insatisfaisant. Effectivement, les règles les plus confiantes correspondent souvent à des motifs peu fréquents ou redondants dans les données explorées et se généralisent difficilement à de nouveaux documents. Dès lors que les règles sont extraites très exhaustivement de données, des cas particuliers (tournures de phrases, expressions linguistiques, etc.) apparaissent, que le système doit utiliser avec précaution. Par ailleurs, les règles moins confiantes pourraient être utilisées, non pour réaliser l'annotation, mais pour conforter une hypothèse d'annotation incertaine. Nous postulons alors la nécessité pour le système de se fonder sur des combinaisons de *preuves concordantes* avant de prendre des décisions. Ainsi, nous sommes amenés à considérer un modèle plus élaboré, dans lequel certaines règles d'annotation constituent individuellement des indices nécessaires (mais non suffisants individuellement) pour réaliser une transduction.

## 6.3 Estimer la vraisemblance des transductions

### 6.3.1 Probabiliser les marqueurs individuellement

C'est alors que nous pouvons directement lier l'utilisation des règles d'annotation au mécanisme de transduction à l'aide de modèles orientés données. Nous remarquons ici que l'utilisation d'une telle approche peut être mise sur le compte de la connaissance imparfaite des entités nommées, tant elles sont variables et revêtent diverses formes. Les organisations, par exemples, sont difficiles à recenser exhaustivement, et souvent ambiguës avec les personnes et les lieux. Comme nous le verrons en section 7.3.2.4, d'autres types sont encore trop exploratoires (marques, produits, événements) pour que l'on puisse en modéliser explicitement la reconnaissance de manière efficace. A notre sens, l'utilisation de modèles numériques nous permet de reconnaître des phénomènes linguistiques dont la

nature est aujourd'hui trop imparfaitement connue.

De plus, nous pouvons maintenant établir un lien direct avec les *instructions* que nous supposons sous-jacentes à la reconnaissance des entités nommées (c.f. 3.3). Effectivement, lorsque l'on tient compte de toutes les règles d'annotation extraites, nous pouvons les interpréter comme des indices contribuant individuellement à l'ajout d'un ou plusieurs marqueurs. Considérons par exemple l'expression complexe suivante :

*'Le <pers> directeur du département d'informatique de l'Université de Tours </pers> enseigne en M2.'*

Il nous semble alors que deux règles d'annotation partielles se focalisant sur un seul marqueur, comme `'DET <pers> NC/directeur'` et `'</pers> VER/enseigner'`, pourraient nous guider efficacement vers l'annotation à réaliser. Il s'agit d'évaluer au fil d'un énoncé les multiples hypothèses de transductions individuelles possibles et d'écarter au fur et à mesure les annotations les moins probables afin de produire en fin de processus l'annotation qui sera jugée la plus vraisemblable. Comme annoncé précédemment, nous adoptons une approche orientée données, mais au lieu de réaliser l'annotation par catégorisation des tokens de l'énoncé, nous modélisons le processus comme insertions de marqueurs (instructions locales) au sein des énoncés.

D'un point de vue formel, considérons une séquence de la base de données enrichie et l'ensemble des règles d'annotation dont nous disposons $\mathcal{R}$. A toute position $j$ de l'énoncé, nous pouvons déterminer quel sous ensemble de règles $\mathcal{R}_j$ propose la transduction d'un ou plusieurs marqueur(s). Nous pouvons ainsi modéliser, pour chaque position de l'énoncé et pour chaque marqueur possible, la probabilité de réaliser la transduction correspondante. Dans un premier temps, les transductions sont dites *atomiques*, car restreintes à l'insertion d'un seul marqueur. Nous la formulons ceci de la manière suivante :

**Probabilité d'une transduction atomique** : soient une séquence de la base de données enrichie $I = i_1 i_2 \dots i_n$ et l'ensemble des règles réalisant des transductions à chaque position de la séquence $\mathcal{R}_1 \mathcal{R}_2 \dots \mathcal{R}_n$, alors la probabilité de réaliser une transduction pour un marqueur $m$ à une position $j$ est définie par $P(m \in M_j | \mathcal{R}_j)$.

Au sein de ce modèle sera également considérée la probabilité de ne réaliser aucune transduction, représentée par un marqueur particulier $\emptyset$ qui sera être estimée manière similaire, $P(M_j = \emptyset | \mathcal{R}_j)$.

### 6.3.2 Inférence bayésienne

La probabilité de réaliser une transduction atomique est alors conditionnée aux règles d'annotations susceptibles de réaliser une transduction. Il nous faut être en mesure d'estimer cette probabilité. Pour cela, nous pouvons utiliser un classifieur bayésien naïf selon la formule suivante :

$$P(m \in M_j | \mathcal{R}_j) = P(m) * \frac{P(\mathcal{R}_j | m \in M_j)}{P(\mathcal{R}_j)}$$

Si l'évaluation des paramètres $P(m)$ et $P(\mathcal{R}_j)$ n'est pas nécessairement problématique, nous ne disposons cependant pas dans le corpus exploré des statistique pour toutes les

combinaisons des associations entre marqueurs et règles $P(\mathcal{R}_j|m \in M_j)$. Pour pallier cela, il est possible de faire une approximation en supposant que les règles d'annotation $T \in \mathcal{R}_j$ sont indépendantes :

$$P(m \in M_j|\mathcal{R}_j) \approx P(m) * \frac{\prod_{T \in \mathcal{R}_j} P(T|m)}{\prod_{T \in \mathcal{R}_j} P(T)}$$

L'avantage de ce modèle est qu'il peut être entièrement construit à partir des statistiques collectées lors de l'exploration du corpus. Cependant, comme nous le verrons lors de nos expériences en partie 9, l'hypothèse d'indépendance entre règles d'annotation trouve rapidement ses limites lorsque le système est alimenté par de nombreuses règles. Nous nous apercevons avoir besoin d'un phase d'ajustement supplémentaire afin de déterminer non seulement comment les règles d'annotation déterminent les marqueurs, mais de surcroît comment elles interagissent.

### 6.3.3 Régression logistique

Comme nous l'avons vu, selon l'ensemble des règles d'annotations dont nous disposons, il peut y avoir de nombreux indices locaux (les règles), potentiellement interdépendants, entrant en jeu dans l'estimation de la probabilité d'un marqueur. La régression logistique, par ajustement de poids au sein d'un modèle exponentiel, nous permet de tenir compte de cette multiplicité de facteurs selon la formule suivante :

$$P(m \in M_j|\mathcal{R}_j) = \frac{exp\left(\sum_{T \in \mathcal{R}_j} \lambda_{T,m}\right)}{Z(\mathcal{R}_j)}$$

Les paramètres $\lambda_{T,m}$ correspondent aux poids des diverses règles dans l'estimation de la probabilité d'un marqueur donné. Le dénominateur $Z(\mathcal{R}_j)$ est un facteur de normalisation. Ce sont ces paramètres qu'il est nécessaire d'ajuster selon une procédure dite d'*apprentissage*, souvent réalisée à l'aide d'un algorithme itératif de descente de gradient [Berger *et al.*, 1996]. Nous n'entrons pas dans le détail de cette méthode de classification mais considérons simplement qu'elle permet de mieux estimer les probabilités des marqueurs selon les règles d'annotation qui se déclenchent.

### 6.3.4 Probabilités de séquences de marqueurs

Il n'est pas suffisant, pour réaliser une annotation, de déterminer quel est le marqueur le plus probable. Même lorsque l'on écarte les imbrications, plusieurs marqueurs peuvent être requis à une position donnée, comme par exemple pour l'annotation *'Le <fonc> président </fonc> <pers> Georges Pompidou </pers>'*. Il nous faut alors être en mesure de faire le lien entre la probabilité d'insérer un marqueur individuel et celle d'insérer une *séquence de marqueurs*. Pour cela, nous tenons compte des probabilités fournies par le modèle et des statistiques issues du corpus concernant les séquences de marqueurs. Nous faisons appel à la règle de Bayes afin de déterminer la probabilité moyenne d'une *séquence de marqueurs* à l'aide de probabilités conditionnelles :

$$P(M_j = m_1 m_2 \dots m_p) = \frac{\sum_{k=1}^{p} P(m_k \in M_k | \mathcal{R}_k) P(m_1 \dots m_p | m_k)}{p}$$

Nous pouvons estimer $P(m_1 \dots m_n | m_k)$, probabilité d'une séquence de marqueurs *conditionnée selon la probabilité d'un marqueur* par les fréquences des séquences de marqueurs rapportées aux fréquences des marqueurs. Les probabilités des séquences de marqueurs calculées doivent de plus être normalisées a posteriori. Remarquons que, lorsque de nombreuses combinaisons de marqueurs sont possibles, ces statistiques pourraient s'avérer insuffisantes. Plus généralement, à cet endroit du modèle peuvent être réalisées diverses optimisations selon les marqueurs dont sont constitués les séquences de marqueurs. Des expériences préliminaires que nous avons pu mener à ce sujet, si elles sortent du cadre de l'approche ici présentée, nous ont montré qu'un travail à ce niveau était une piste prometteuse pour améliorer les performances globales.

## 6.4 Des séquences de marqueurs à l'annotation

Enfin, lorsque les probabilités de séquences de marqueurs $P(M_i)$ sont estimées, il convient de les utiliser afin de déterminer quelle annotation est la plus vraisemblable *parmi les annotations valides*. A cet effet, nous faisons une hypothèse d'indépendance entre marqueurs insérés au sein d'un énoncé :

$$P(M_1 M_2 \dots M_n) \approx \prod_{j=1}^{n} P(M_j)$$

Parmi les marqueurs qu'il est possible d'insérer aux diverses positions d'un énoncé, un nombre restreint de combinaisons forment une annotation valide selon le guide. Généralement, outre le fait qu'une entité nommée est nécessairement reconnue par deux marqueurs (ouvrants et fermants) et que les chevauchements sont interdits, il peut aussi s'agir de stipuler quelles imbrications sont tolérées. Dans les cas que nous aurons à prendre en compte, il sera aisé de déterminer si une séquence de marqueurs $M_j$ à insérer à la position $j$, lorsque l'on connaît les marqueurs précédemment positionnés $M_1 \dots M_{j-1}$, amorce une annotation qui pourra être valide. L'espace de recherche pourra ainsi être exploré en examinant séquentiellement les marqueurs à insérer. Rechercher les annotations valides les plus vraisemblables sera résolu grâce à des techniques de programmation dynamique que nous exposerons en section 9.1.

Pour illustrer l'ensemble du processus, nous présentons le mécanisme de recherche d'une annotation potentiellement récursive pour l'énoncé *'Il visite le Centre Georges Pompidou'*. L'enrichissement à l'aide de la morpho-syntaxe et des lexiques (dont les célébrités `'CELEB'` et les bâtiments `'BAT'`) construit une séquence de la base de données enrichie formée d'autant d'items (avec ambiguïté dans les lexiques signalée par la disjonction exclusive $\oplus$). De plus, admettons que nous disposions, pour cet exemple minimal, de trois règles parmi les règles d'annotation extraites (exprimées en motifs de segments) candidates à réaliser des transductions sur cette séquence :

- $R_1$ : `'VERB/visiter DET <loc> BAT </loc>'`,
- $R_2$ : `'DET/le <loc> BAT/NP/Centre'`,
- $R_3$ : `'<pers> CELEB/NP </pers>'`.

La figure 6.1 indique où sont réalisées les transductions possibles à partir des règles qui se déclenchent. Nous y mentionnons également les probabilités estimées de la manière suivante :

- $P($ <loc> $|R_1, R_2) = 0.8$ et $P(\emptyset|R_1, R_2) = 0.2)$,
- $P($ <pers> $|R_3) = 0.7$ et $P(\emptyset|R_3) = 0.3)$,
- $P($ </loc> $|R_1, R_3) = 0.5$, $P($ </pers> $|R_1, R_3) = 0.6$ et $P(\emptyset|R_1, R_3) = 0.1)$.

Nous indiquons les probabilités des séquences de marqueurs après normalisation, ainsi que les trois possibilités d'annotations à considérer. Par ordre de vraisemblance croissante, il est possible de ne rien annoter, d'annoter une personne, un lieu, ou une personne imbriquée dans un lieu. Si le schéma d'annotation l'autorise, l'annotation préférée contiendra donc une personne comme composante d'un lieu, ce qui semble plausible. Si l'imbrication n'est pas tolérée, le lieu sera plus vraisemblable, grâce aux indices que constituent les deux règles $R_1$ et $R_2$.

La complexité du système réside donc, comme souvent pour ce type de tâche, dans l'élaboration du modèle plutôt que dans son application. L'extraction des règles d'annotation nous fournit des éléments suffisamment riches pour ne pas chercher à tenir compte de manière sophistiquée des dépendances entre marqueurs au sein d'un énoncé. Nous restons dans une perspective selon laquelle les *instructions* construisent séparément le début et la fin des entités, sans avoir à en reconnaître nécessairement tous les tokens. Alors que la majorité des modèles orientés données sont contraints à modéliser les dépendances entre tokens à catégoriser (HMM, CRF), nous ne nous préoccupons pas de cet aspect et, au lieu de cela, explorons exhaustivement et simultanément l'*ontologie* et la *structure* des données en lien avec les marqueurs afin de structurer les énoncés à l'aide d'indices locaux apportés par les règles d'annotation. La construction de l'annotation à l'aide de ces indices est réalisée plus tardivement.

Nous avons ici présenté la manière dont sont exploitées les données, d'une part pour extraire des règles d'annotation, d'autre part pour paramétrer un modèle qui nous permet de déterminer les annotations les plus vraisemblables pour un énoncé donné selon les règles extraites. Il s'agit maintenant de confronter ce modèle à la tâche qui nous occupe, qui consiste non seulement à reconnaître des entités nommées, mais aussi à structurer (reconnaissance avec imbrications) un texte en entités nommées.

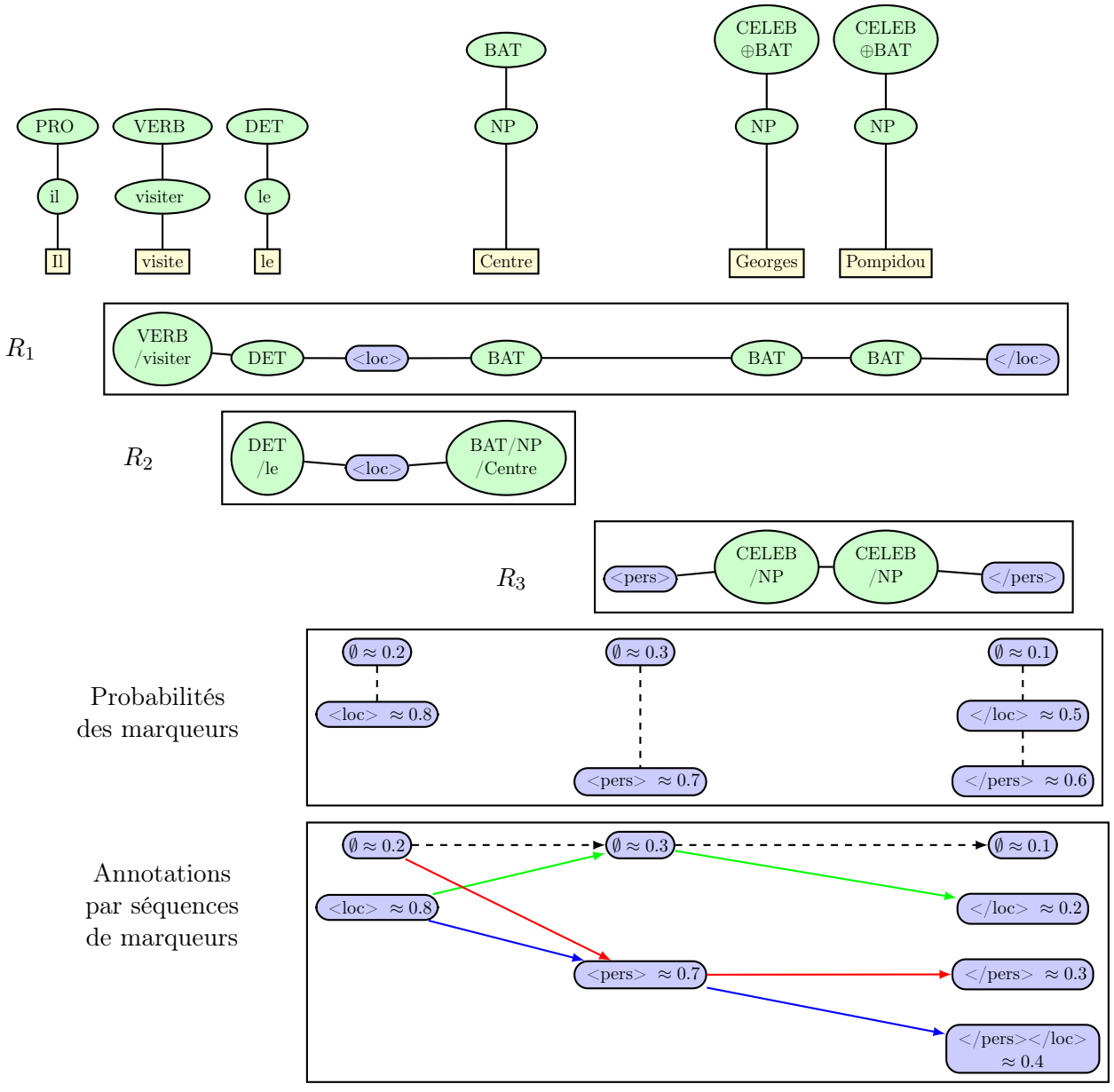



FIGURE 6.1 – Probabilité des marqueurs selon les règles d'annotation

Troisième partie

# mXS : extraction de règles d'annotation pour structurer

# Chapitre 7

# Cadre expérimental

## 7.1 Architecture générale

Nous présentons maintenant l'implémentation et l'évaluation de l'approche que nous avons proposée dans les parties précédentes. De manière générale, le processus que nous cherchons à mettre en œuvre peut être vu comme un *apprentissage automatique* [Mitchell, 1997] et comporte donc deux phases distinctes. La première étape est le *paramétrage* du modèle à partir de données pour lesquelles les entités nommées sont connues, soit l'extraction des règles d'annotation et l'estimation des paramètres d'un modèle numérique utilisant ces règles, comme décrits en section 6.3. La seconde étape utilise ces paramètres au sein d'un système pour réaliser une *prédiction*, dans notre cas une *annotation*, de données pour lesquelles nous cherchons à reconnaître les entités nommées. En faisant abstraction des prétraitements, ce principe est résumé en figure 7.1.

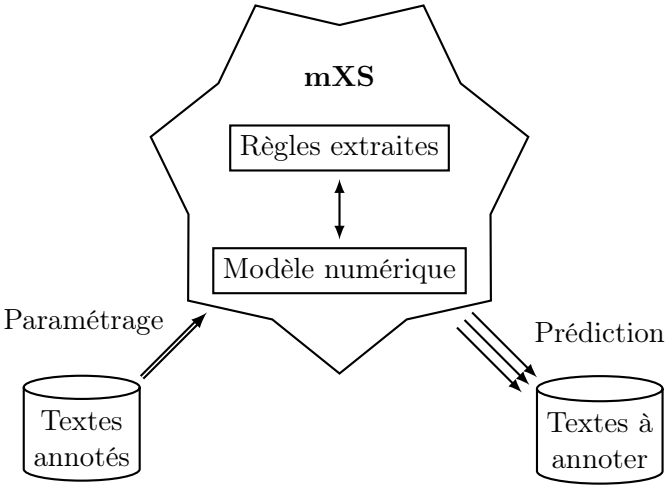



FIGURE 7.1 – Paramétrage et prédiction de mXS

Une différence notable entre ces deux étapes est que la première nécessite des passes (une pour chaque niveau) sur *toutes les données* afin d'extraire les règles (selon leur fréquence

et leur confiance comme mentionné en section 5.5) et d'ajuster les paramètres du modèle numérique (par exemple les probabilités pour le calcul de la vraisemblance comme formulé en section 6.3), tandis que la prédiction peut être réalisée séparément sur chaque énoncé, donc en une seule passe sur des parties des données.

Quelle que soit l'étape considérée, nous réalisons les mêmes enrichissements sur les données à l'aide d'analyses préalables (c.f. 5.2). Comme le système doit être en mesure de traiter de la parole spontanée, sans connaissance a priori du type de texte (modalité, thème) fourni en entrée, nous sommes contraints de mettre en œuvre des traitements fiables, robustes et peu profonds : tokenisation, lemmatisation, catégorisation en morpho-syntaxe, traits sémantiques. De manière schématique, nous concevons notre système selon une architecture analogue aux chaînes de traitements, comme l'illustre la figure 7.2 :

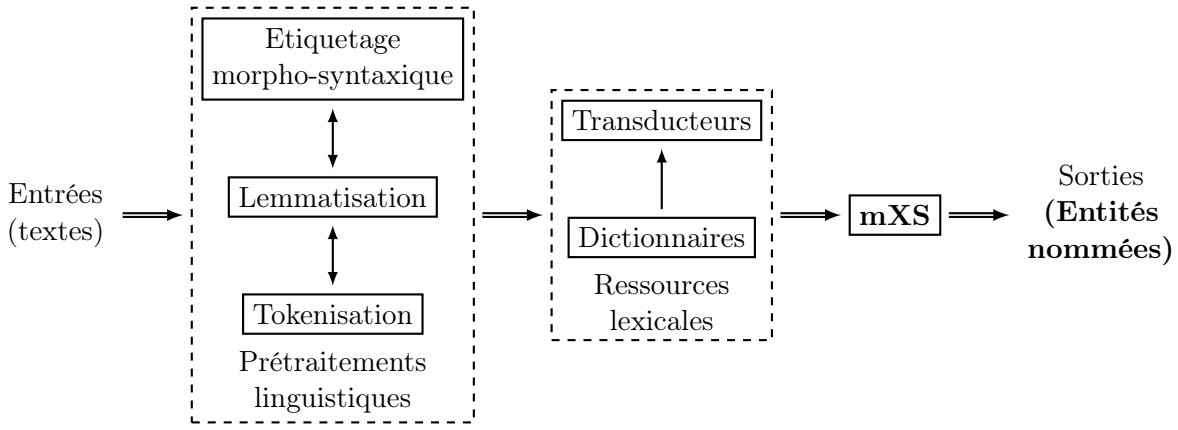



FIGURE 7.2 – Architecture des traitements

Nous disposons d'outils libres qui réalisent efficacement les prétraitements linguistiques, notamment TreeTagger [Schmid, 1994]. Notre travail portera donc essentiellement sur la bonne exploitation des *ressources lexicales* et sur l'implémentation du module `mXS`. Si le procédé présenté par la figure 7.2 est le même pour le paramétrage ou pour la prédiction, les données en entrée, en sortie et les fonctionnalités de `mXS` activées sont :

- **Paramétrage** :
  - *entrées* : textes annotés en entités nommées,
  - `mXS` : extraction de règles et estimation des paramètres numériques,
  - *sorties* : règles d'annotation et modèle numérique.
- **Prédiction** :
  - *entrées* : textes bruts,
  - `mXS` : application des règles et du modèle numérique,
  - *sorties* : textes annotés en entités nommées.

Ainsi, les prétraitements linguistiques et l'utilisation de ressources lexicales fonctionnent de manière identique, modulo les annotations en entités nommées, qui sont présentes lors du paramétrage mais évidemment absentes lors de la prédiction. L'enrichissement des données est réalisé à l'aide de connaissances (linguistiques, lexicales) que l'on suppose utiles pour analyser le langage et en particulier pour reconnaître les entités nommées. Ces connais-

sances mises à disposition doivent ainsi permettre à `mXS` de réaliser son paramétrage lors de l'exploration des données de manière à ce que que la phase de prédiction reste performante sur d'autres données.

## 7.2 Modules de traitements et ressources

### 7.2.1 Tokenisation, lemmatisation, étiquetage morpho-syntaxique

Pour réaliser les prétraitements linguistiques, après essai de divers outils disponibles (MElt, Macaon, FRMG), nous utilisons TreeTagger [Schmid, 1994] qui a l'avantage d'être couramment utilisé par la communauté TAL et d'être relativement robuste aux divers types de textes fournis en entrée. Par ailleurs, cet outil réalise conjointement la tokenisation, la lemmatisation et l'étiquetage morpho-syntaxique. Sur l'énoncé *'Je suis au Centre Georges Pompidou.'*, TreeTagger nous fournit une sortie comme l'indique le tableau 7.1.

| **Token** | **Morpho-syntaxe** | **Lemme** |
|---|---|---|
| Je | `'PRO:PER'` | je |
| suis | `'VER:pres'` | suivre\|être |
| au | `'PRP:det'` | au |
| Centre | `'NAM'` | <unknown> |
| Georges | `'NAM'` | Georges |
| Pompidou | `'NAM'` | Pompidou |
| . | `'SENT'` | . |

TABLE 7.1 – Exemple de sortie TreeTagger

A partir d'une séquence de caractères, TreeTagger crée une ligne par token détecté. Chaque ligne est formée de trois colonnes, séparées par des tabulations. La première contient le token dans sa forme originale, la seconde sa catégorisation morpho-syntaxique, la dernière son lemme. Nous utilisons ces informations pour segmenter le texte en tokens puis pour procéder aux premiers enrichissements. Remarquons à ce sujet que le token *'suis'*, volontairement ambigu, donne lieu à deux hypothèses de lemmatisation séparées par la barre verticale `'|'`, cette ambiguïté est représentée à l'aide de la disjonction exclusive. Par ailleurs, l'utilisation des deux points `':'` correspond à une sous-catégorisation en morpho-syntaxe, dont nous pouvons tenir compte. Tel quel, nous obtiendrions l'énoncé enrichi suivant :

```
'PRO/PER/je/Je VER/pres/suivre/suis⊕VER/pres/être/suis PRP/det/au/au
NAM/<unknown>/Centre NAM/Georges/Georges NAM/Pompidou/Pompidou SENT/./.'
```

Les catégories morpho-syntaxiques de TreeTagger sont résumées dans le tableau 7.2, auxquelles nous adjoignons pour information les proportions relevées au sein du corpus Etape (c.f. 7.3.3). Cependant, cet outil n'ayant pas vocation à traiter les entités nommées en particulier, nous procédons à quelques adaptations afin de nous focaliser sur les informations que nous supposons déterminantes pour reconnaître les entités nommées et de ne pas tenir compte de celles qui nous paraissent moins utiles. En voici la description détaillée :

– **Verbes** : les sous-catégories relatives au mode et temps du verbe sont supprimées.

| Nom | Catégorie | Sous-catégories | Proportion |
|---|---|---|---|
| ABR | abréviation | | 0,18% |
| ADJ | Adjectif | | 4,09% |
| ADV | Adverbe | | 8,25% |
| DET | Déterminant | ART, POS | 8,11% |
| INT | Interjection | | 0,53% |
| KON | Conjonction | | 5,24% |
| NAM | Nom propre | | 3,34% |
| NOM | Nom commun | | 14,07% |
| NUM | Nombre | | 1,15% |
| PRO | Pronom | DEM, IND, PER, POS, REL | 15,81% |
| PRP | Préposition | det | 11,36% |
| PUN | Ponctuation, hors fin d'énoncés | cit | 7,32% |
| SENT | Token de fin d'énoncé | | 4,64% |
| SYM | Symbole non identifié | | 0,05% |
| VER | Verbe | cond, futu, impe, impf, infi, pper, ppre, pres, simp, subi, subp | 15,87% |

TABLE 7.2 – Catégories morpho-syntaxiques de TreeTagger

- **Noms propres et abréviations** : ces deux catégories sont rattachées à une catégorie plus générale `'NAMABR'`.
- **Prépositions** : la sous-catégorie `'PRP:det'` (*'au'*, *'du'*, *'aux'*, *'des'*) forme une catégorie à part `'PRPDET'`.
- **Déterminants** : les déterminants définis (*'le'*, *'la'*, *'les'*, *'l"*) sont sous-catégorisés en `'DET/DEF'`.
- **Nombres** : les nombres sont sous-catégorisés selon le nombre de chiffres dont ils sont formés, ce nombre étant précisé s'il est inférieur ou égal à quatre (`'NUM/DIGITS:MANY'`, `'NUM/DIGITS:4'`, `'NUM/DIGITS:3'`, `'NUM/DIGITS:2'`, `'NUM/DIGITS:1'`) et, dans ce dernier cas, le préfixe des deux premiers chiffres sous-catégorise de nouveau (`'NUM/DIGITS:4/PREF:19'`, `'NUM/DIGITS:4/PREF:20'`).
- **Nom propres, abréviations, noms, verbes** : ces éléments sont sous-catégorisés selon le suffixe des trois derniers caractères du lemme (`'NOM/SUFF:ier'`, `'NOM/SUFF:eur'`, `'NAM/SUFF:ges'`, `'VER/SUFF:vre'`).
- **Noms propres** : pour TreeTagger, une lemmatisation des noms propres correspond à une recherche dans ses dictionnaires (s'il est trouvé le token est affiché, sinon `'<unknown>'`), nous ne tenons pas compte de cette information.

Ces diverses adaptations ont été apportées au fur et à mesure de l'implémentation du système et ont été testées afin d'améliorer la reconnaissance des entités nommées (lorsque des informations sont ajoutées) ou de réduire la richesse des données à explorer (lorsque des informations sont supprimées). Nous notons cependant que la plupart d'entre elles ne modifient pas de manière décisive les performances du système, sauf celles concernant les

nombres qui permettent de distinguer les dates d'autres expressions numériques (les années sont formées de quatre chiffres et commencent souvent par *'19'* ou *'20'*).

Suite à ces adaptations, l'énoncé présenté ci-dessus sera alors enrichi en :

```
'PRO/PER/je/Je VER/SUFF:vre/suivre/suis⊕VER/SUFF:tre/être/suis PRPDET/au/au
NAMABR/NAM/SUFF:tre/Centre NAMABR/NAM/SUFF:ges/Georges
NAMABR/NAM/SUFF:dou/Pompidou SENT/./.'
```

Ces enrichissements nous fournissent les premiers niveaux de généralisation disponibles lors de l'exploration des données et la recherche de règles d'annotation. Ils reposent sur des analyses de nature linguistique et ne portent pas sur une catégorisation sémantique de tokens ou d'expressions composées. Cependant, comme ils s'appuient sur la morphologie des tokens (morpho-syntaxe, nombres, suffixes), ces informations ont pour objectif d'apporter une certaine robustesse au système, en particulier en ce qui concerne la reconnaissance d'expressions qui n'ont jamais été observées telles quelles lors de l'exploration de données et qui ne seraient pas présentes dans les ressources lexicales.

Par ailleurs, lors de l'exploration des données, notre objectif est d'extraire des règles d'annotation relativement génériques. Pour ce faire, nous émettons l'hypothèse que les variations de flexion (déclinaisons, conjugaisons) sont peu pertinentes lors de la recherche de motifs corrélés aux entités nommées. En conséquence, la phase d'exploration des données, pour le tokens qui ne sont pas des noms propres (dont le traitement particulier est détaillé en section 7.2.2), nous effaçons les tokens eux-même pour ne conserver que le lemme. Dans l'exemple précédent, nous obtenons alors pour explorer les données :

```
'PRO/PER/je VER/SUFF:vre/suivre⊕VER/SUFF:tre/être PRPDET/au
NAMABR/NAM/SUFF:tre/Centre NAMABR/NAM/SUFF:ges/Georges
NAMABR/NAM/SUFF:dou/Pompidou SENT/.'
```

De manière générale, nous faisons remarquer la flexibilité avec laquelle il est possible de réaliser des enrichissements plus ou moins profonds selon les tokens considérés. Effectivement, n'ayant pas de contraintes a priori sur la profondeur ou la largeur des hiérarchies que nous explorons pour extraire les règles d'annotation, nous pouvons nous permettre de moduler à volonté l'axe ontologique selon les éléments observés et la tâche d'annotation à réaliser.

### 7.2.2 Ressources lexicales

Si les traitements linguistiques nous permettent d'extraire des règles d'annotation qui détectent et reconnaissent certaines entités nommées, nous constatons la nécessité d'affiner nos enrichissements selon la sémantique que portent certaines expressions linguistiques. Pour illustrer cette nécessité, prenons en considération les deux énoncés *'Ce 3 juillet était maudit.'* et *'Ce 3 pièces était maudit.'*. Nous voyons qu'en l'occurrence, une règle d'annotation telle que `'<date> PRO/DEM/ce NUM NOM </date>'` conduirait à une reconnaissance erronée dans le second énoncé. Il faut espérer qu'une règle telle que `'NOM/SUFF:let/juillet </date>'` fasse la distinction entre ces deux énoncés. Mais alors, une telle règle serait nécessaire pour chaque mois de l'année. A la place de quoi, nous pouvons remplacer `'NOM/SUFF:let/juillet'` par une catégorie sémantique, en l'occurrence `'MOIS'`, qui serait partie d'une règle telle que `'<date> PRO/DEM/ce NUM MOIS </date>'`.

Ce principe peut être étendu à de nombreuses expressions linguistiques (en particulier les noms propres) que l'on peut alors regrouper selon des catégories sémantiques. Comme nous l'avons mentionné en 3.1 et en 3.2, de nombreuses approches reposent sur cette possibilité de généraliser des expressions linguistiques en des traits sémantiques. En conséquence, de nombreuses ressources ont été développées à cet effet (par exemple WordNet ou Prolex). Lorsque nous explorons les données, ceci présente deux avantages : certaines tournures peuvent devenir suffisamment fréquentes grâce à ces généralisations pour être extraites sous forme de règles d'annotation et nous évitons alors d'extraire plusieurs règles lexicalisées si elles ont une généralisation commune.

Pour ce faire, nous exploitons des ressources externes diverses (dont certaines sont importées à partir des dictionnaires et motifs du système CasEN, présenté ci-après en section 7.2.3) que nous manipulons sous forme de dictionnaires. Nous les fusionnons avec quelques listes, qui ont été constituées manuellement, selon les performances constatées du système, pour les fonctions, lieux, organisations, quantités et dates. Nous obtenons une base lexicale comme l'indique la figure 7.3.

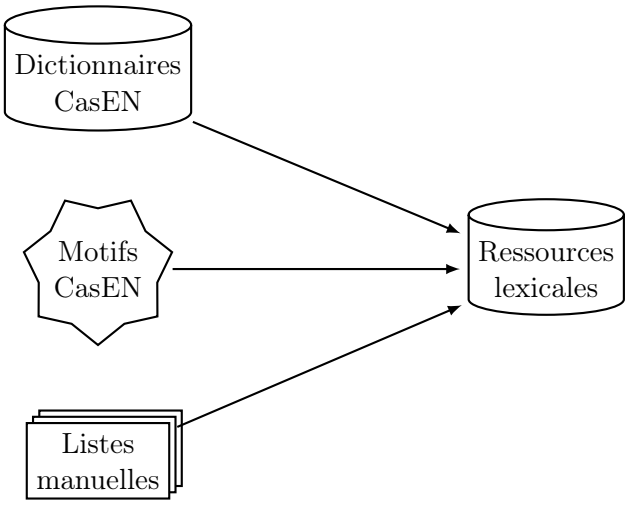



FIGURE 7.3 – Ressources lexicales

Au total, ces ressources contiennent 221 547 expressions distinctes qui produisent 443 112 catégorisations sémantiques (il est fréquent que plusieurs catégories sémantiques soient associées aux entrées). Nous donnons en figure 7.3 la liste des catégories ayant plus de 9 entrées (dont en gras celles qui en concernent plus de 1000), réparties arbitrairement par types d'entités nommées. Nous voyons que de nombreuses entrées sont dédiées à la reconnaissance des personnes et des lieux, tandis qu'il y en a comparativement moins pour les organisations. Notons également qu'une partie de ces ressources sont générées à partir d'automates (certains transducteurs CasEN) qui peuvent s'avérer très productifs, comme par exemple `listMouvement`.

Nous sommes en mesure d'utiliser ces ressources pour enrichir les données sur lesquelles les règles s'appuient. Nous les utilisons telles quelles pour produire un nouvel enrichissement, qui pourra dans de nombreux cas être sémantiquement ambigu. Ainsi, l'énoncé *'En*

| **Entité** | **Catégories sémantiques** |
|---|---|
| Personnes | ANIM (497), DYN (57), ETHNO (398), **IND (71140)**, listCiviliteNobiliaire (30), listFamille (153), listPersIndividuel (96), listTitre (9), listTitreNobiliaire (41), listTitreReligieux (43), **PREN (28456)**, SURNOM (77) |
| Lieux | ASTRO (35), BAT (293), GEO (386), **HYDRO (5458)**, listBatiment (10), listBatimentReligieux (111), listBatimentVille (10), listBoutique (9), listDebutNomVille (15), listEcole (211), listEntreprise (11), listHopital (934), listHotel (39), listNomRue (22), LOC-DIR (16), LOC-PHY-PRE (23), **NAT (2289)**, **REG (5756)**, SNAT (252), **TER (8489)**, **TOPO (128144)**, **VILLE (113521)** |
| Organisations | ASSO (540), **COLL (2406)**, **COM (1809)**, ENT (726), INST (86), listAdjectifParti (16), listClub (32), listCommerceEtranger (335), listLoisir (68), **listMouvement (28095)**, listNomMinistere (59), listNomMinistereMaj (10), listNomMinistereOrgAdm (18), listOrganisationMinDe (71), listOrganisationPre (61), listOrgCommerceGauche (25), listPersCollective (133), RELIG (151), SPORT (298), **MEDIA (1350)**, **ORG (1859)**, ORG-ENT (22), ORG-INST (81), ORG-INST-IN (10), ORG-INST-PRE (12), ORG-LOC-GOV (62), ORG-MED (89), ORG-NICK (20), ORG-POL (44), ORG-POL-IN (19), ORG-SPORT-AFF (15), ORG-SPORT-LOC (48), POL (208) |
| Temps | EVT (604), listAdjectifDate (12), MON (511), TIME-DAY (9), TIME-EVENT (22), TIME-MOD-PRE (37), TIME-MON (13), TIME-QUANT (24), TIME-REL (22) |
| Montants | listLongueurSurface (32), listLongueurVolume (24), listMasse (10), listSurface (14), QUANT-MOD (67), QUANT-UNIT (105) |
| Produits | listDocument (9), listOeuvre (13), OBJ (16), **PROD (2924)** |
| Fonctions | JOB-COMP (626), JOB-PRE (72), listArtisan (10), listArtiste (40), listFoncCollective (92), listFoncGouvernement (62), listFoncMilitaire (124), listSpecialiteMedicale (93), listSportif (11), **PRO (31585)** |

TABLE 7.3 – Catégories sémantiques

*1970, Pompidou a été à Washington'* sera enrichi de la manière suivante (nous factorisons les suffixes de l'opérateur ⊕ pour plus de lisibilité) :

```
'TIME-MOD-PRE/PRP/en/En NUM/DIGITS:4/PREF:19/1970/1970 PUN/,/,
IND⊕TOPO⊕POL⊕VILLE/NAMABR/NAM/SUFF:dou/Pompidou
VER/SUFF:oir/avoir/a VER/SUFF:tre/être/été PRP/à/à
IND⊕TOPO⊕ORG-LOC-GOV⊕PREN⊕VILLE/NAMABR/NAM/SUFF:ton/Washington'
```

A l'image des variations flexionnelles dont nous faisons abstraction lors de l'exploration des données (c.f. 7.2.1), nous considérons que les noms propres, étant une classe ouverte, n'ont pas vocation à être utilisés au sein des règles d'annotation. Une fois que les noms propres ont donné lieu à des enrichissements sémantiques, nous omettons les items lexicaux eux-même pour extraire des règles d'annotation qui ne reposent que sur les catégories sémantiques. Ainsi, l'exemple précédent deviendra, pour l'exploration des données :

```
'TIME-MOD-PRE/PRP/en/En NUM/DIGITS:4/PREF:19/1970/1970 PUN/,/,
IND⊕TOPO⊕POL⊕VILLE/NAMABR/NAM/SUFF:dou
VER/SUFF:oir/avoir/a VER/SUFF:tre/être/été PRP/à/à
IND⊕TOPO⊕ORG-LOC-GOV⊕PREN⊕VILLE/NAMABR/NAM/SUFF:ton'
```

Les noms propres *'Pompidou'* et *'Washington'*, qui peuvent être associées à de nombreuses entités hors-contexte, ont reçu tous les enrichissement sémantiques dont nous avons connaissance. Nous alimentons ainsi l'axe ontologique par des informations que l'exploration de données sera en charge de sélectionner (à l'aide du ⊕) lors de l'extraction des règles d'annotation. Si, en l'occurrence, ces traits sémantiques sont insérées comme un seul niveau de la hiérarchie, il reste possible d'imaginer des enrichissements plus sophistiqués, à partir de bases de connaissances structurées, ou d'informations issues d'autres traitements (syntaxe, catégorisation verbale, coréférence, etc.).

### 7.2.3 CasEN : système à base de connaissances

Comme nous l'avons mentionné en section 3.3, notre travail s'inspire conjointement des approches orientées connaissances et orientées données. Dans ce cadre, nous nous situons dans la continuité des travaux de Friburger [Friburger, 2002, Friburger, 2006] et disposons en particulier de la cascade `CasEN` (librement distribuée [1] avec le module de cascade de transducteurs Cassys pour la chaîne de traitement Unitex). En plus d'exploiter des ressources communes pour enrichir les données (c.f. 7.2.2), nous utilisons ce système orienté connaissances de reconnaissance d'entités nommées.

Initialement conçu pour reconnaître des noms propres sur des textes écrits par reconnaissance de preuves contextuelles externes ou internes [Friburger, 2002, Friburger, 2006], le système `CasEN` est constitué d'un ensemble de transducteurs qui s'appuient sur des dictionnaires à large couverture (dont Prolex [Tran et Maurel, 2006, Bouchou et Maurel, 2008]) pour reconnaître les entités nommées incrémentalement [Friburger et Maurel, 2004, Friburger et Maurel, 2011, Maurel *et al.*, 2011]. Implémenté par introspection, ce système a été régulièrement évalué et amélioré afin de s'adapter au traitement de la parole et à divers schémas d'annotation. Nous décrivons ici ce système dans les grandes lignes, tel qu'il a été utilisé dans le cadre de la campagne d'évaluation Etape (c.f. 7.3.3).

1. A l'adresse `http://tln.li.univ-tours.fr/Tln_CasEN.html`

Les dictionnaires mis à disposition des transducteurs contiennent au total 888 561 expressions distinctes, réparties au sein de quatre dictionnaires comme le décrit le tableau 7.4. Pour chaque entrée, de nombreux traits peuvent être définis, qui peuvent être morphologiques, morpho-syntaxiques ou sémantiques. Tous ne sont pas dédiés à la reconnaissance d'entités nommées, notamment le dictionnaire Delaf conçu pour le TAL sans visée applicative particulière.

| **Dictionnaire** | **Contenu** | **Taille** |
|---|---|---|
| Delaf | Formes fléchies (hors noms propres) | 682 418 |
| Prolex | Noms propres et dérivés | 118 309 |
| Dico `CasEN` | Noms propres complémentaires, description définies | 87 408 |
| Dico `CasEN` ambiguïtés | Noms propres particulièrement ambigus | 426 |

TABLE 7.4 – Dictionnaires CasEN

Le système `CasEN` repose sur le principe d'une cascade : des transducteurs sont appliqués sur les textes selon un ordre prédéfini. Ces transducteurs peuvent faire appel à d'autres transducteurs en leur sein, afin de détecter des indices locaux ou reconnaître des expressions imbriquées. Ils peuvent également tirer parti d'annotations précédemment établies au sein de la cascade. Au total, `CasEN` comporte 790 transducteurs implémentés manuellement. Parmi ceux-ci, la cascade fait directement appel à 100 d'entre eux. Nous voyons dès lors que le système est élaboré selon une récursivité importante, pour partie lié à la nécessité de factoriser certaines reconnaissances, en particulier les expressions qu'il est utile de détecter pour plusieurs types d'entités nommées (assimilées dans le projet Etape aux *composants*).

La figure 7.4 illustre ce mode de fonctionnement. Nous voyons que sa complexité rend difficile une visualisation globale des dépendances entre transducteurs. A gauche de la figure se trouvent les types principaux d'entités nommées à reconnaître. Au centre apparaissent, dans l'ordre, les 100 transducteurs dont est constituée la cascade. Sur la partie droite se trouvent, dans un ordre arbitraire, les autres transducteurs auxquels il est fait appel. Les arcs tracés entre les types d'entités nommées et les transducteurs de la cascade indiquent les reconnaissances réalisées par les transducteurs (y compris par appel de transducteurs). Les arcs entre les transducteurs au centre et ceux de droite correspondent aux appels de transducteurs. Notons que nous n'y affichons pas les dépendances entre transducteurs au sein du groupe du milieu ou du groupe de droite, ni l'utilisation d'annotations établies par d'autres transducteurs au cours de la cascade.

Deux mécanismes permettent donc de construire la représentation incrémentale. D'une part, un transducteur peut faire appel à d'autres transducteurs en son sein, ils réalisent alors les reconnaissances simultanément. D'autre part, une annotation établie par un transducteur de la cascade est disponible pour les transducteurs ultérieurs au sein de la cascade. Précisons à ce sujet que le système ne permet pas de modifier une annotation établie par un transducteur de la cascade, seulement de l'imbriquer à l'intérieur d'une autre annotation. De ce point de vue, les reconnaissance réalisées par un transducteur de la cascade ne peuvent être remises en cause. Il ne s'agit donc pas d'émettre des hypothèses entre lesquelles prendre une décision en fin de cascade, mais d'ordonner correctement les transducteurs afin que chaque reconnaissance réalisée soit correcte.

L'appel de transducteurs est illustré en figure 7.5 par un transducteur dédié à la re-

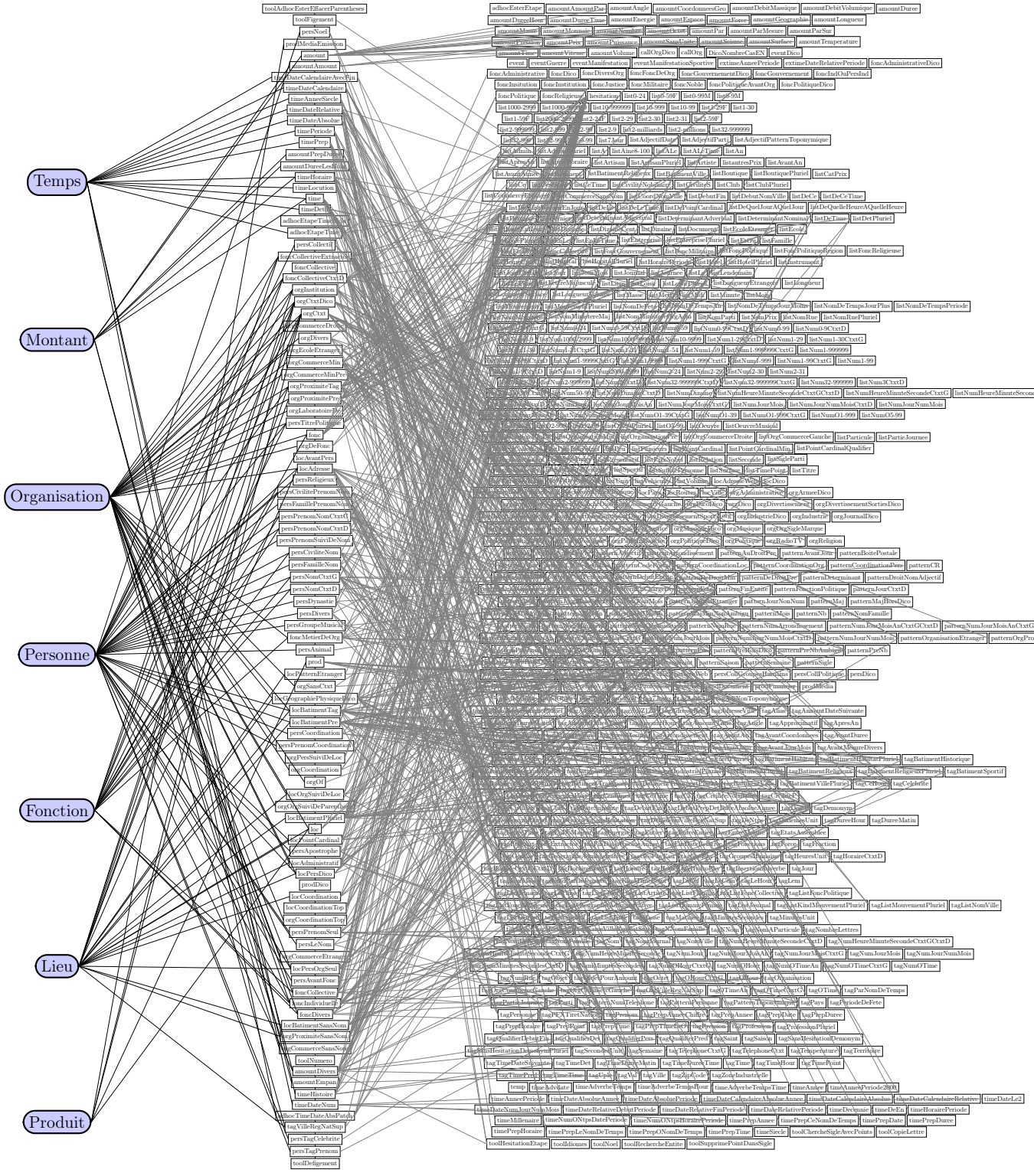
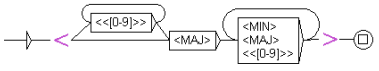


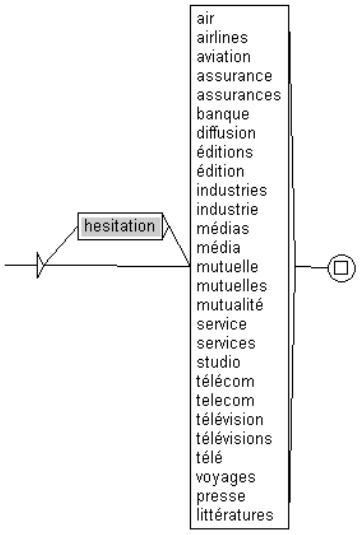

FIGURE 7.4 – Graphe des transducteurs

connaissance de noms d'organisations commerciales comme *'France-Presse'* qui sollicite le transducteur morphologique générique `patternPreNbAmbigu` (potentiellement utilisable par d'autres transducteurs) et au transducteur contextuel spécifique aux organisations `listOrgCommerceDroite`. L'utilisation des reconnaissances au sein de la cascade est présenté en figure 7.6, avec un transducteur qui reconnaît des fonctions lorsqu'un trait d'organisation commerciale (`'N+Commerce'`) est disponible, et de même manière des personnes à l'aide de fonctions (`'func'`) préalablement reconnues. Ces exemples nous permettent d'illustrer les nombreuses possibilités offertes par utilisation de transducteurs lors de l'implémentation d'un système orienté connaissances.


(a) Organisation commerciale




(b) Morphologie de noms d'organisation `patternPreNbAmbigu`



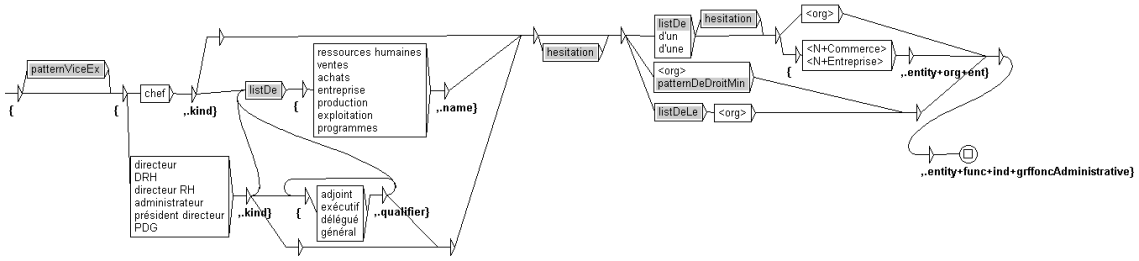

(c) Liste de suffixes d'organisations `listOrgCommerceDroite`

FIGURE 7.5 – Appels de transducteurs

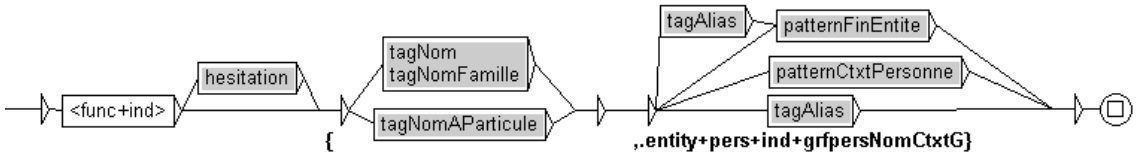
Pour illustrer en pratique le fonctionnement des transducteurs, considérons l'énoncé suivant :

*'Le musée Georges Pompidou est inauguré le 31 janvier 1977.'*

La première analyse réalisée par la chaîne de traitement Unitex comporte une tokenisation et une application de dictionnaires qui fournissent des informations morphologiques, morpho-syntaxiques ou lexicales. La richesse des dictionnaires Unitex (dont locutions et expressions composées) donne lieu à de nombreuses analyses possibles dès ces premières étapes. Pour prendre en compte les diverses hypothèses d'analyses, la repré-




(a) Fonction administrative au sein d'une organisation commerciale




(b) Personne reconnue avec sa fonction administrative

FIGURE 7.6 – Cascade de transducteurs

sentation construite prend la forme d'un *DAG* (Directed Acyclic Graph [2]), comme nous l'illustrons en figure 7.7. Outre les ambiguïtés concernant les formes, nous y notons une tokenisation fine des chiffres *'31'* et *'1977'* ainsi que les nombreux traits sémantiques affectés à *'Georges Pompidou'* à l'aide des dictionnaires (`'Hum'`, `'Anthroponyme'`, `'Celebrite'`, mais aussi `'Toponyme'`, `'Ville'`).

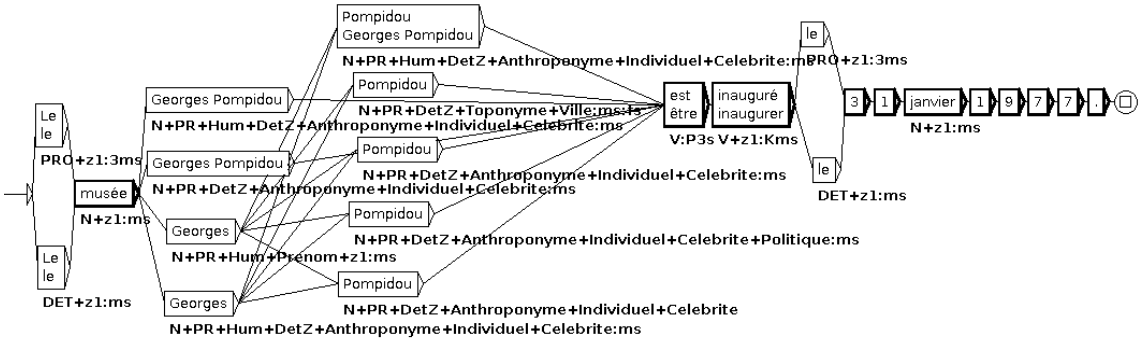



FIGURE 7.7 – Texte représenté sous forme de DAG

Le transducteur affiché en figure 7.8 reconnaitra, à l'aide du trait sémantique `'Celebrite'` et du préfixe *'musée'*, une organisation de divertissement nommée d'après une célébrité. Nous y remarquons en particulier l'appel au transducteur `'tagCelebrite'`, lui-même chargé de reconnaître une célébrité d'après les traits sémantiques. Dans son ensemble, nous voyons que le transducteur réalise la description du langage naturel selon un modèle génératif (c.f. 1.3). Par ailleurs, nous notons que la sortie générée par le transducteur peut fournir, en plus des entités nommées, des informations supplémentaires sur les transducteurs qui ont

2. graphe dirigé sans cycles

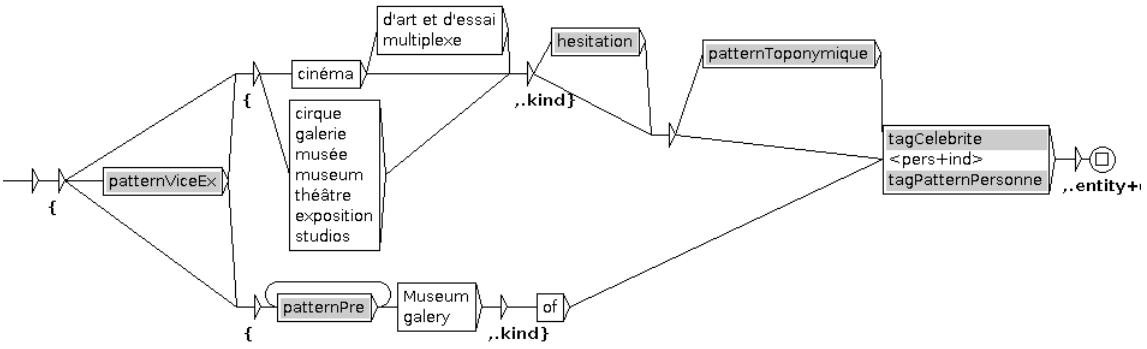



FIGURE 7.8 – Transducteur reconnaissant les organisations de divertissement

participé à la reconnaissance d'une expression. L'utilisation de ce transducteur parmi bien d'autres nous conduira à obtenir l'étiquetage suivant en entités nommées :

```
'Le { {musée,.kind} {Georges,.name+first+grftagPrenom}
{Pompidou,.name+last+grftagNNomFamille} ,.entity+org+ent+grforgDivertissementSortiesDico}
est inauguré { {le,.time-modifier} {31,.day} {janvier,.month} {1977,.year}
,.entity+time+date+abs+grftimeDateCalendaireAbsolueAnnee}.'
```

Si ce système s'avère précis dans la description du langage naturel, son implémentation et son adaptation sur des tâches diverses sont généralement assez longues. Comme souvent, l'objectif de ne pas commettre d'erreurs dans les reconnaissances d'entités nommées (précision) doit être concilié avec la nécessité de reconnaître toutes les entités nommées (rappel). Généralement, les automates implémentés cherchent à reconnaître *tous* les tokens des expressions qui forment des entités nommées à l'aide de divers indices. Comme annoncé en section 3.3, nous remettons en cause ce principe et cherchons à déterminer s'il est possible de ne reconnaître que les marqueurs (frontières, balises) d'entités nommées, sans avoir à en décrire systématiquement le contenu. Notons que ceci ne semble pas contradictoire avec l'utilisation de mécanismes de transduction.

Enfin, nous remarquons que de nombreux transducteurs parmi ceux qui sont appelés (partie droite de 7.4), réalisant la reconnaissance de motifs partiel, peuvent générer des expressions utiles à la reconnaissance d'entités nommées. Cette connaissance peut ainsi être exploitée comme une ressource lexicale (c.f. 7.2.2) qui fournit des indices d'ordre sémantique afin de réaliser nos enrichissements. Une partie de ces transducteurs a ainsi été sélectionnée afin d'alimenter les ressources lexicales de `mXS`.

## 7.3 Jeux de données

### 7.3.1 Campagnes d'évaluation en entités nommées en France

Le travail dont nous rendons compte ici a été réalisé dans le contexte des campagnes d'évaluations françaises en traitement de la parole. Notre travail démarre à la suite de la campagne Ester2 [3] (Evaluation des Systèmes de Transcription Enrichie d'Émissions Radio-

3. `http://www.afcp-parole.org/camp_eval_systemes_transcription/`

phoniques, 2008-2009), pour se conclure avec la campagne Etape [4] (Evaluations en Traitement Automatique de la Parole, 2011-2012), cette dernière campagne ayant de nombreuses interactions avec le programme de recherche Quaero [5] (2008-2013).

A l'image du cycle de campagnes MUC aux États-Unis, ces projets de recherche, coordonnés par l'AFCP [6] (Association Francophone de la Communication Parlée), réunissent des acteurs scientifiques et industriels autour de tâches dédiées au traitement automatique de la parole (c.f. 2.2.2). La spécificité d'Ester2 et d'Etape est de se focaliser sur le traitement de données issues d'émissions radiodiffusées et télévisuelles, donc orales et, en partie, spontanées. A partir d'enregistrements audio, des traitements sont appliqués, afin d'obtenir des informations exploitables pour diverses visées applicatives, dont la recherche et l'extraction d'information. Le processus est essentiellement constitué des tâches suivantes :

- **Segmentation** : le signal audio est segmenté et ses segments sont qualifiés (locuteur, locuteurs multiples, bruits de fond, musiques, etc.).
- **Transcription** : les portions du signal audio qui contiennent de la parole sont transformées en texte (reconnaissance automatique de la parole).
- **Recherche d'information** : des informations (les entités nommées) sont annotées au sein des textes transcrits.

A chaque tâche définie, une donnée de référence est manuellement constituée. Celle-ci permet, d'une part, d'analyser la problématique et d'élaborer des systèmes, d'autre part, d'évaluer les participants. Dans le cadre d'un processus modulaire, il est alors possible de mesurer la performance obtenue par chaque module séparément (en supposant que les modules précédents ont parfaitement fonctionné) ou l'ensemble des traitements (avec possibilité de bruits ou d'erreurs en sortie des modules de segmentation ou de transcription). Nous présentons ici la tâche de recherche d'information des campagnes Ester2 et Etape, dont les corpus nous ont permis de mener nos expériences en reconnaissance d'entités nommées.

### 7.3.2 Corpus Ester2

#### 7.3.2.1 Schéma d'Annotation

La campagne Ester2 s'inspire des nombreux projets et travaux à l'international (c.f. 2.4.3) pour spécifier la manière dont sont annotées les entités nommées. Le système `CasEN` a participé à cette campagne d'évaluation pour la recherche d'information. Le guide d'annotation, intitulé *'Entités nommées, dates, heures et montants'* [7], définit comme suit les éléments à reconnaître :

*'Les entités nommées sont au cœur de la problématique de l'extraction de l'information d'un document [...] Les entités nommées sont des types particuliers d'unités lexicales (groupes de mots) qui font référence à une entité du monde concret [...] et qui ont un nom [...] Une EN a généralement une existence relativement stable dans le temps [...] Notons*

4. `http://www.afcp-parole.org/etape.html`
5. `http://www.quaero.org`
6. `http://www.afcp-parole.org`
7. `http://www.afcp-parole.org/camp_eval_systemes_transcription/docs/Conventions_EN_ESTER2_v01.pdf`

*que les informations "temps" et "montant" ne sont pas des entités nommées mais qu'elles sont visées par les tâches d'extraction d'information. L'ensemble peut être appelé entités spécifiques.'*

Comme nous le remarquons en section 2.2.2, le guide mentionne dans cet avant-propos la visée opératoire (*'extraction de l'information'*) et une forme de stabilité de la désignation (*'référence à une entité du monde concret'*). La proposition que nous faisons en 2.5 nous paraît alors couvrir les *'entités spécifiques'*, tout en l'élargissant, car nous ne faisons pas appel aux entités du *'monde concret'* ni à un besoin applicatif spécifique. Plus en détail, le guide d'annotation Ester2 décrit les types d'entités nommées à considérer de la manière suivante :

| | |
|---|---|
| **Personne** (`pers`) | Humain réel ou fictif, animal réel ou fictif |
| **Fonction** (`fonc`) | Politique, militaire, administrative, religieuse, aristocratique |
| **Organisation** (`org`) | Politique, éducative, commerciale, non-commerciale, médiatique et de divertissement, géo-socio-administrative |
| **Lieu** (`loc`) | Géographique naturel, région administrative, axe de circulation, adresse (postale, téléphone et fax, électronique), construction humaine |
| **Production humaine** (`prod`) | Moyen de transport, récompense, œuvre artistique, production documentaire |
| **Date et heure** (`time`) | Date (absolue, relative), heure |
| **Montant** (`amount`) | Age, durée, température, longueur, surface et aire, volume, poids, vitesse, autre, valeur monétaire |

TABLE 7.5 – Types d'entités nommées Ester2

Cette typologie couvre, dans l'essentiel, les entités que nous mentionnons en 2.3. Les entités nommées sont annotées en contexte, ce qui signifie qu'il faut notamment tenir compte des métonymies. A l'usage, certains sous-types sont particulièrement difficiles à distinguer pour des systèmes automatiques (ou parfois même pour l'humain), notamment les *organisations géo-socio-administrative* des *régions administratives (lieux)* ou les *dates et heures relatives* des *durées (montants).*

Au sujet des imbrications, le guide d'annotation donne les directives suivantes :

- *"Annoter les entités nommées de lieu, de fonction, d'organisation et de personne lorsque celles-ci sont imbriquées. Annoter la mention de l'entité nommée la plus large possible."*
- *"Ne pas identifier toute entité nommée imbriquée dans une entité nommée se référant à une adresse postale ou électronique."*
- *"Ne pas identifier les entités nommées de personne à l'intérieur des autres types d'entité (organisation, lieu...). Ceci afin d'éviter toute confusion entre la personne et l'entité ciblée."*
- *"Annoter une fonction même si celle-ci est imbriquée dans une entité nommée pers.hum."*

Ainsi, l'imbrication d'annotations est volontairement limitée (dans les données, elle concernera approximativement 11% des annotations). De manière générale, la campagne Ester2 étant la première initiative de grande ampleur dans un contexte francophone, ses objectifs sont plus ambitieux sur la quantité de données à mettre à disposition que sur la profondeur et la finesse des annotations réalisées. De fait, les sous-types ne seront pas

considérés lors des évaluations, seuls les types principaux seront pris en compte. Pour les détails des conventions adoptées au sein du guide d'annotation, nous renvoyons au site de l'AFCP [8].

#### 7.3.2.2 Données

Le corpus Ester2 que nous utilisons pour nos expériences est issu de transcriptions manuelles d'émissions radiodiffusées. Ces émissions correspondent à des émissions d'information, des dossiers d'actualités, mais aussi, plus marginalement, quelques discussions et des débats. Les radios enregistrées sont francophones, de régions géographiques françaises ou d'Afrique du nord (France Inter, France Info, France Classique, France Culture, RFI, RTM, TVME, Africa1). Selon les besoins de la campagne, ce corpus est scindé en plusieurs parties. La table 7.6 donne le nom et le volume de données pour chaque partie du corpus (les énoncés étant segmentés selon l'étiquette morpho-syntaxique `'SENT'` de TreeTagger) : `Ester2-Train` (entraînement des systèmes à apprentissage), `Ester2-Dev` (développement des systèmes), `Ester2-Test` (évaluation des systèmes).

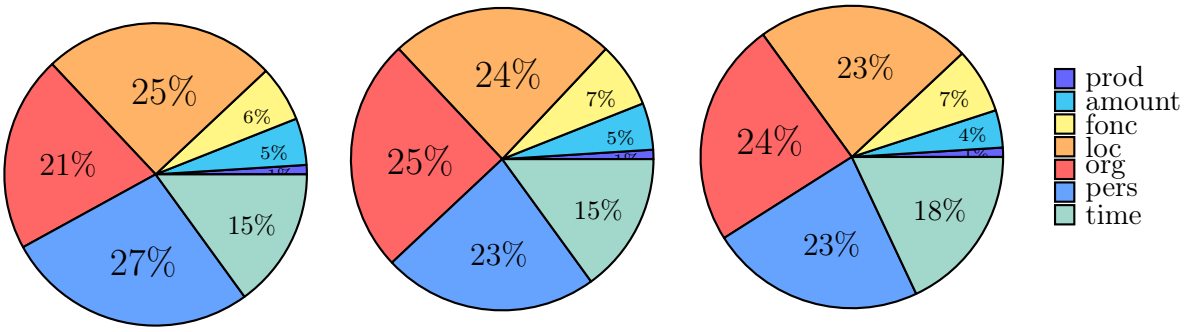
| Corpus | Source (nombre de fichiers) | Tokens | Énoncés | EN |
|---|---|---|---|---|
| **Ester2-Train** | France Inter (47), RFI (29), RTM (103), France Info (13), France Culture (1), France Classique (1) | 1 269 327 | 44 211 | 80 227 |
| **Ester2-Dev** | Africa1 (9), France Inter (5), RFI (2), TVME (4) | 73 386 | 2 491 | 5 326 |
| **Ester2-Test** | Africa1 (9), France Inter (6), RFI (7), TVME (4) | 87 165 | 2 983 | 5 875 |
| **Total** | 211 enregistrements | 1 429 878 | 49 685 | 91 428 |

Table 7.6 – Caractéristiques pour chaque partie d'Ester2

Dans cette table, la colonne *Source* indique le nombre de fichiers par radio source, chaque fichier correspondant à une émission enregistrée (qui peuvent être de durées très diverses). Nous remarquons la disparité des radios utilisées pour constituer la partie `Ester2-Train` par rapport à `Ester2-Dev` et `Ester2-Test`. Les graphiques de la figure 7.9 montrent la répartition des types d'entités nommées au sein de ces parties du corpus Ester2. Nous y constatons une relative homogénéité des proportions à travers les parties du corpus. Globalement, le nombre d'entités nommées rapporté au nombre de tokens du corpus est de 6,4%. Nous notons la très faible proportion de `prod` ($\approx 1\%$), et l'équilibre entre `pers`, `loc` et `org` qui réunissent une large part ($\approx 70\%$) des entités nommées à reconnaître.

Constituer un corpus d'une telle taille est un travail remarquable. Comme nous l'indiquons en 2.4.1, l'annotation manuelle de corpus est une tâche complexe, il est encore difficile d'évaluer objectivement la qualité d'une annotation produite manuellement. Cependant, de l'avis de nombreux participants, les annotations sur le corpus sont d'une qualité contestable, et ceci plus particulièrement pour la partie `Ester2-Train`, très volumineuse, difficilement exploitable en l'état et nécessitant de nombreuses corrections ou adaptations [Raymond et Fayolle, 2010]. Au cours de nos travaux, les quelques expériences que nous avons me-

8. `http://www.afcp-parole.org/camp_eval_systemes_transcription/docs.html`




(a) Ester2-Train (b) Ester2-Dev (c) Ester2-Test

FIGURE 7.9 – Répartition des types d'entités nommées pour chaque partie d'Ester2

nées avec cette partie des données ont dégradé les performances globales du système, nous écartons donc cette partie du corpus lors de nos expériences.

#### 7.3.2.3 Métriques d'évaluation et performances des systèmes

L'évaluation de la performance des systèmes (dont `CasEN`) au cours de la campagne Ester2 est réalisée par calcul du SER (c.f. 2.4.2), en confrontant les sorties des systèmes et les annotations manuelles. A cet effet, le corpus `Ester2-Test`, dont les annotations manuelles ne sont pas connues des participants, est utilisé. En plus du SER, la précision et le rappel des systèmes sont indiqués pour information. Enfin, notons qu'une phase d'adjudication a eu lieu, qui a permis de corriger en partie les annotations de référence, mais à laquelle l'équipe ayant développé `CasEN` n'a pas eu le temps de participer.

| Part. | **Manuel** | | | **LIMSI** (12,11 WER) | | | **LIA** (17,83 WER) | | | **IRISA** (26,09 WER) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S | P | R | S | P | R | S | P | R | S | P | R |
| `LIA` | 23,9 | 86,46 | 71,85 | 43,4 | 79,52 | 59,45 | 51,6 | 76,51 | 55,02 | 56,8 | 72,26 | 49,02 |
| `LIMSI` | 30,9 | 81,15 | 70,94 | 45,3 | 75,13 | 62,33 | 55,5 | 70,50 | 57,52 | 61,2 | 66,13 | 50,67 |
| `LINA` | 37,1 | 80,75 | 55,48 | 54,0 | 71,98 | 44,01 | 60,4 | 68,76 | 40,84 | 65,2 | 63,66 | 35,66 |
| `LI Tours` | 33,7 | 79,39 | 65,82 | 50,7 | 71,36 | 54,16 | 80,8 | 56,59 | 46,46 | 82,9 | 51,28 | 42,38 |
| `LSIS` | 35,0 | 82,65 | 73,07 | 55,3 | 70,23 | 58,39 | 86,5 | 70,36 | 28,66 | 88,6 | 67,03 | 25,22 |
| `Synapse` | 9,9 | 93,02 | 89,37 | 44,9 | 76,39 | 67,16 | 60,7 | 70,26 | 59,21 | 66,2 | 65,95 | 52,71 |
| `Xerox` | 9,8 | 93,61 | 91,50 | 44,6 | 58,91 | 70,06 | na | na | na | na | na | na |

TABLE 7.7 – SER (S), Précision (P) et Rappel (R) des systèmes en reconnaissance des entités nommées, campagne Ester2

Nous reproduisons en table 7.7 les performances obtenues par les participants lors de la campagne Ester2, sur les transcriptions manuelles et sur les transcriptions issues de systèmes de reconnaissance automatique de la parole (`LIMSI`, `LIA` et `IRISA`) [Galliano *et al.*, 2009]. Le système `CasEN`, *LI Tours*, y obtient la cinquième place sur les transcriptions manuelles. Nous voyons que tous les systèmes ont une précision plus élevée que leur rappel. Les deux premiers systèmes (`Xerox` et `Synapse`) sont orientés connaissances, tandis que le troisième (`LIA`) est orienté données. Nous observons que les performances se dégradent nettement avec la qualité des transcriptions automatiques pour les approches orientées

connaissance. A contrario, le système du LIA soit tolérant aux erreurs introduites dans les transcriptions automatiques. Il semble ainsi que les systèmes orientés données soient plus tolérants au bruit.

Nous remarquons également la variabilité des évaluations selon que l'on considère le SER, la précision et le rappel. Par exemple, le système du LSIS obtient une meilleure précision et un meilleur rappel que ceux du LIMSI et de CasEN, mais son SER le classe moins bien. Le SER étant mesuré par pondération d'erreurs, un hypothèse plausible est que le premier commet plus d'erreurs d'insertion ou de délétion, tandis que les deux autres commettent en contrepartie plutôt des erreurs d'extension ou de type. Comme remarqué en 2.4.2, la mesure pondère les erreurs de telle sorte que les entités nommées ont intérêt à être détectées, même lorsqu'elles sont mal reconnues.

#### 7.3.2.4 Analyse du comportement de CasEN pour Ester2

Nous avons procédé à une analyse plus détaillée des performances obtenues par CasEN lors de la campagne Ester2. En premier lieu, nous avons sélectionné des fichiers représentatifs au sein de Ester2-Test. Sur ces données, nous avons comptabilisé les erreurs, et cherché à les caractériser ou à corriger les annotations manuelles s'il y a lieu. Cette sous-partie d'Ester2-Test contient 12 fichiers, 39 707 tokens et 2 792 entités nommées, nous la nommons Ester2-Corr. Le détail des évaluations de CasEN pour chacun de ces fichiers et pour l'ensemble d'Ester2-Corr est donné en table 7.8 (les erreurs I, D, T E entrent en compte pour le calcul du SER, par simplification, nous n'indiquons pas les erreurs multiples). Les autres fichiers d'Ester2-Test forment la sous-partie Ester2-Held (47 458 tokens, 3 083 entités nommées).

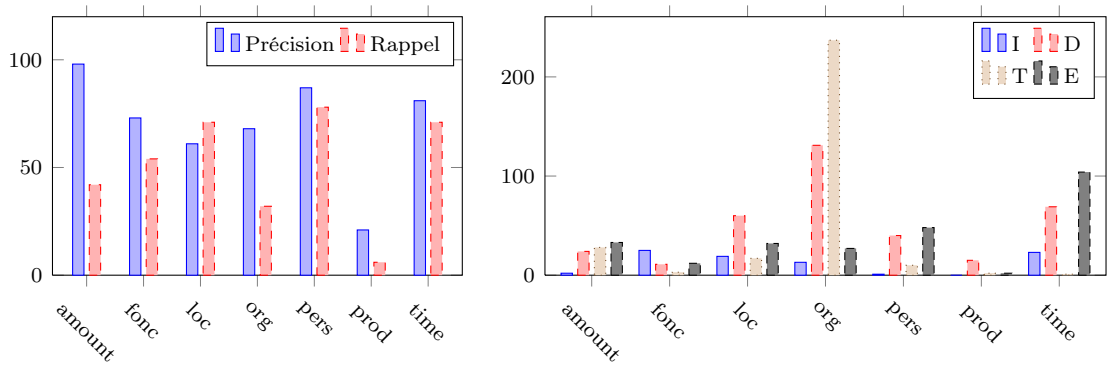
| Fichier | EN | I | D | T | E | SER |
|---|---|---|---|---|---|---|
| 20080122_0930_0940_RFI | 133 | 2 | 19 | 16 | 11 | 31,20 |
| 20080207_1200_1210_AFRICA1 | 170 | 2 | 19 | 24 | 11 | 27,59 |
| 20071221_1900_1920_INTER | 318 | 11 | 41 | 29 | 36 | 31,19 |
| 20080125_2030_2040_RFI | 151 | 5 | 16 | 15 | 12 | 29,80 |
| 20080118_1000_1100_INTER | 331 | 21 | 74 | 42 | 26 | 43,63 |
| 20080107_2135_2150_TVME | 216 | 9 | 28 | 39 | 30 | 38,61 |
| 20080122_2030_2040_RFI | 177 | 5 | 21 | 25 | 29 | 34,69 |
| 20080124_2030_2040_RFI | 151 | 2 | 16 | 14 | 12 | 26,09 |
| 20080108_2135_2150_TVME | 176 | 3 | 41 | 23 | 20 | 40,00 |
| 20071218_1900_1920_INTER | 329 | 15 | 41 | 27 | 42 | 33,47 |
| 20080123_2030_2040_RFI | 162 | 1 | 15 | 17 | 15 | 27,96 |
| 20080118_2030_2040_RFI | 165 | 7 | 19 | 27 | 14 | 34,97 |
| **Total** | 2479 | 83 | 350 | 298 | 258 | 34,07 |

TABLE 7.8 – Détail des erreurs de d'Insertion(I), de Délétion (D), de Type (T) et d'Extension (E) du système CasEN sur les fichiers d'Ester2-Corr

Nous y constatons d'assez fortes disparités dans les performances globales (SER) selon le fichier analysé. La difficulté du système à reconnaître les entités nommées semble liée au type d'émission considérée, ce qui remet en cause la robustesse du système face à des données orales, parfois spontanées, voire bruitées. Précisons cependant que le système,

dédié à la reconnaissance des entités nommées à l'écrit, n'a pas été adapté à l'analyse de transcriptions de parole. Par ordre d'importance, le système commet des erreurs de délétion, de type, d'extension puis d'insertion.

L'examen de la précision, du rappel et des erreurs détaillées par types d'entités nommées, comme montré par le graphique 7.10, nous permet de mieux cibler les catégories concernées par ces erreurs. Nous voyons effectivement que le type `org`, présent en quantité importante dans le corpus, correspond à de nombreuses erreurs de délétion et de type qui le pénalisent fortement. Ceci est notamment lié à la présence de noms propres de pays interprétées par métonymie à des équipes sportives (`'La <org> France </org> a battu l'<org> Allemagne </org> au mondial.'`) qui n'ont pas été reconnues par le système. Nous remarquons également la présence de nombreuses erreurs d'extension sur les `time` en partie liées à l'annotation d'article et de prépositions (`'<time> le 10 juillet </time>'`, `'<time> en 1968 </time>'`, etc.).




(a) Précision et rappel

(b) Insertion(I), de Délétion (D), de Type (T) et d'Extension (E)

FIGURE 7.10 – Performance de `CasEN` par type d'entités nommées, campagne Ester2

Sur ces fichiers, nous examinons plus en détail les 1 197 erreurs relevées lors de l'évaluation. Au cours de ce travail, nous avons été amenés à comptabiliser 99 erreurs portant sur des annotations qui nous paraissaient contestables selon les règles édictées dans le guide d'annotation (types des entités nommées, extensions). Nous avons corrigé en conséquence l'annotation manuelle de référence. La performance du système est alors évaluée à 31,0 SER (-2,7) [Nouvel *et al.*, 2010a], sans préjuger par ailleurs des gains que les autres systèmes participant à cette campagne auraient obtenus sur les données après ces corrections. Surtout, nous avons ainsi obtenu sur la sous-partie `Ester2-Corr` du corpus une annotation que nous jugeons de bonne qualité.

De nombreux autres constats sont dressés suite à cette analyse d'erreurs, qui nous permettent de guider les améliorations à apporter à `CasEN` [Nouvel *et al.*, 2010a, Nouvel *et al.*, 2010b] et, plus généralement, les directions à considérer dans l'optique d'une exploration automatique des données.

En particulier, nous avons constaté que, parmi les erreurs d'extension, 88 concernent la frontière droite des entités nommées, 222 leur frontière gauche (certaines concernant les deux simultanément). Cette statistique montre que, contrairement à ce que l'intuition pour-

rait laisser penser, le système a des difficultés à déterminer le *début* des entités nommées. En extrapolant, nous supposons que `CasEN` parvient difficilement à décrire l'intégralité d'expressions complexes résultant de composition, comme par exemple *'le directeur du département d'informatique de l'Université de Tours'*. Notre proposition de reconnaissance d'entités nommées par recherche de marqueurs présentés en section 3.3 vise à trouver une solution nouvelle en déterminant les frontières des entités nommées sans nécessairement en reconnaître tout le contenu. Nous espérons, de ce fait, parvenir à élaborer un système de reconnaissance d'entités nommées robuste, tout en conservant une mécanique similaire aux transducteurs.

### 7.3.3 Corpus Etape

#### 7.3.3.1 Schéma d'Annotation

La spécification du schéma d'annotation pour le projet Etape est réalisé par le laboratoire LIMSI, en coordination avec le projet Quaero [9]. Il s'inspire très largement de ce qui a été défini lors du projet Ester2, tout en étendant les annotations en diversité (types) et en profondeur (niveaux de description). Une des nouvelles pistes de recherche étudiée à l'occasion de ce projet concerne la capacité des systèmes à décrire les éléments qui constituent les entités nommées. Ces éléments sont appelés *composants* et doivent être annotés par les participants au même titre que les entités nommées.

Nous rappelons (et enrichissons) les éléments donnés par le guide d'annotation [10] concernant les entités nommées et leurs composants :

*"les entités nommées incluent traditionnellement trois grandes classes : les noms, les quantités, les dates et durées. Nous nous plaçons dans le contexte d'extraction d'information (entités, relations) servant à constituer une base de connaissances. Les entités nommées étendues forment le pivot de la base de connaissances [...] Nous étendons donc la définition habituelle des entités nommées, centrée sur les noms propres, à des expressions construites autour de noms communs [...] Les entités annotées ont vocation à être ultérieurement reliées par des relation [...] Une entité est formée d'un ou plusieurs composants, ainsi éventuellement que des parties non annotées."*

Ces travaux inspirent la définition des entités nommées que nous proposons en section 2.5. Comme précédemment, nous retenons la visée applicative et opératoire associée aux entités nommés. Par ailleurs, nous faisons le lien entre la *structuration* des entités nommées en composants et les *instructions* locales sous jacentes liées aux marqueurs d'annotation (c.f. 3.3). La table 7.9 reporte les types d'entités nommées, leurs descriptions et leurs sous-types tels que décrits dans le guide d'annotation. Nous notons que les entités nommées sont étendues à des expressions construites à partir de noms communs, ce qui amène à considérer une plus large gamme d'expressions linguistiques.

Nous relevons l'apparition du type `event`, mais celui-ci restera relativement peu annoté dans les données. La distinction entre les types `loc` et `org` est plus claire que pour Ester2, les *organisations géo-socio-administratives* étant cette fois-ci catégorisées en lieux (admi-

9. http://www.quaero.org
10. http://quaero.org/media/files/bibliographie/quaero-guide-annotation-2011.pdf

| | | |
|---|---|---|
| **Personne** (`pers`) | On distingue deux sous-types de personnes, les individus et les groupes. | ind, coll |
| **Fonction** (`fonc`) | Métier (un pompier), fonction (les chefs d'État, Miss Italie), rôle social (des opposants), etc., d'une personne (func.ind) ou d'un ensemble de personnes (func.coll). | ind, coll |
| **Organisation** (`org`) | On ne définit pas de type organisation <org> général, mais directement des sous-types <org.ent>, <org.adm>. Une organisation peut être exprimée par un genre (<kind>) et des précisions (<name>, <demonym>, <loc>...) | ent, adm |
| **Lieu** (`loc`) | Localisations, lieux, entités spatiales. | adm (town, reg, nat, sup), phys (geo, hydro, astro), oro, fac, add (phys, elec), other |
| **Production humaine** (`prod`) | Cette catégorie est proche de la catégorie PRODUCT de Sekine. | object, art, media, fin, soft, award, serv, doctr, rule, other |
| **Point dans le temps** (`time`) | Nous distinguons dates (time.date) et heures (time.hour). À l'intérieur de ces deux types, nous distinguons également temps absolu (.abs) et temps relatif (.rel). | date (abs, rel), hour (abs, rel) |
| **Quantités** (`amount`) | Une quantité est composée d'une valeur assortie éventuellement d'une unité de mesure, suivie éventuellement d'un objet. | |
| **Evénements** (`event`) | Il est gênant dans l'annotation de ne pas pouvoir annoter les événements (par exemple, le 23e congrès de la CFDT). Leur absence fait se demander comment annoter, avec le guide actuel, des entités qui sont des événements. On introduit donc une définition souple des événements. | |

TABLE 7.9 – Types d'entités nommées Etape

nistratifs). Nous constatons, notamment pour les organisations, fonctions et montants un moins grand nombre de catégories, l'objectif étant d'évaluer le système sur les catégories les plus fines. Comme pour Ester2, les entités nommées sont annotées en contexte.

| | |
|---|---|
| **Transverses** | name (nickname), kind, extractor, demonym (nickname), qualifier, val, unit, range-mark |
| **Personne** | name (last, first, middle) |
| **Adresse** | address-number, po-box, zip-code |
| **Quantité** | object |
| **Temps** | day, week, month, year, century, millenium, reference-era, time-modifier |

TABLE 7.10 – Types de composants Etape

En plus des entités nommées, leurs *composants* sont annotés, dont nous reportons la liste issue du guide d'annotation en table 7.10. Ces nouveaux éléments à annoter sont imbriqués au sein des entités nommées. Sauf quelques exceptions (name, object, unit), les composants ne peuvent eux-même contenir d'autres annotations. De plus, l'imbrication concerne également les entités, comme l'indique le guide d'annotation :

*'Une entité peut servir de composant dans une autre entité [...] A priori, nous excluons les imbrications récursives : une entité ne peut pas être composant d'une entité du même sous-type [...] La hiérarchie des types de composants inclut donc la hiérarchie des entités.'*

Par rapport à Ester2, les annotations seront alors bien plus profondes, à la fois par la possibilité d'imbriquer des entités nommées et par l'obligation d'annoter les composants des entités nommées. Ces nouvelles perspectives demandent à adapter les systèmes par rapport à Ester2, en particulier les approches orientées données qui réalisent une classification de tokens ou qui fonctionnent avec un format BIO (c.f. 3.2). Désormais, une expression linguistique peut simultanément être un composant et un constituant d'une ou plusieurs entités nommées de plus large empan. L'approche par marqueurs (c.f. 3.3) que nous défendons nous paraît en mesure d'appréhender cette problématique d'annotation et de structuration de manière unifiée.

#### 7.3.3.2 Données

Le corpus Etape comporte des enregistrements d'émissions télévisuelles, dont le signal audio est extrait et utilisé pour des tâches similaires à celles définies dans le cadre d'Ester2. Ce sont également des émissions d'information, des dossiers d'actualités, des discussions et des débats. Nous postulons que ces données sont a priori au moins aussi difficiles à traiter que les données du corpus Ester2. Comme pour Ester2, les données sont scindées en trois parties, `Etape-Train` (entraînement des systèmes à apprentissage), `Etape-Dev` (développement des systèmes) et `Etape-Test` (évaluation des systèmes), dont nous donnons les caractéristiques en table 7.11.

De plus, en coordination entre le projet Etape et le programme Quaero, des données issues de la campagne Ester2 ont été annotées en entités nommées selon le nouveau schéma d'annotation et mises à disposition des participants. Nous n'en faisons pas usage mais en mentionnons les caractéristiques à part, pour information, en table 7.11 (`Etape-Quaero`).

Contrairement à Ester2, nous notons pour Etape la grande cohérence des sources uti-

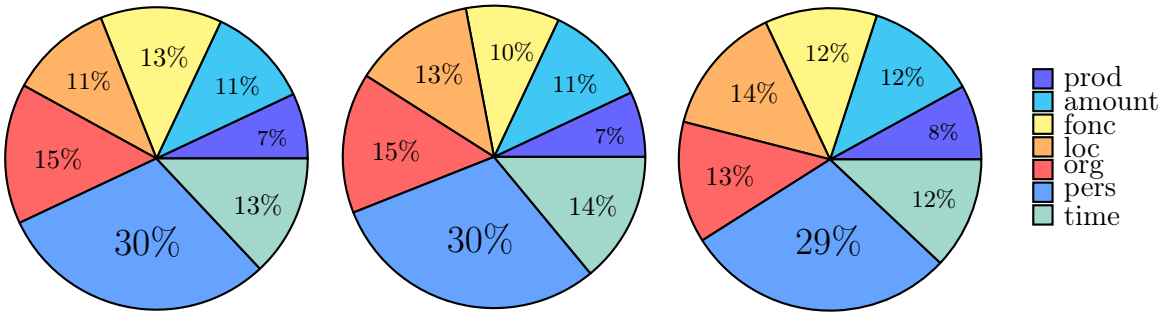
| Corpus | Sources(nombre de fichiers) | Tokens | Énoncés | EN |
|---|---|---|---|---|
| **Etape-Train** | BFMTV (5), France Inter (16), LCP (23) | 355 975 | 14 989 | 46 259 |
| **Etape-Dev** | BFMTV (1), France Inter (6), LCP(6), TV8 (2) | 115 530 | 5 724 | 14 112 |
| **Etape-Test** | BFMTV (1), France Inter (6), LCP (5), TV8 (2) | 123 221 | 6 770 | 13 055 |
| **Total** | 74 enregistrements | 594 726 | 27 483 | 73 426 |
| **Etape-Quaero** | France Classique (1), France Culture (1), France Inter (62), France Info (13), RFI (14), RTM (97) | 1 596 427 | 43 828 | 279 797 |

TABLE 7.11 – Caractéristiques pour chaque partie d'Etape

| Type | NB | % | Type | NB | % |
|---|---|---|---|---|---|
| **name** | 10133 | 13,8 | **week** | 225 | 0,31 |
| **kind** | 7759 | 10,57 | **time.hour.abs** | 224 | 0,31 |
| **pers.ind** | 7364 | 10,03 | **prod.fin** | 209 | 0,28 |
| **name.last** | 5272 | 7,18 | **loc.adm.sup** | 197 | 0,27 |
| **name.first** | 5268 | 7,17 | **day** | 175 | 0,24 |
| **val** | 3931 | 5,35 | **name.nickname** | 150 | 0,2 |
| **amount** | 3413 | 4,65 | **prod.object** | 144 | 0,2 |
| **org.ent** | 2439 | 3,32 | **prod.rule** | 134 | 0,18 |
| **time-modifier** | 2222 | 3,03 | **loc.phys.geo** | 109 | 0,15 |
| **func.ind** | 2157 | 2,94 | **loc.add.elec** | 82 | 0,11 |
| **unit** | 2120 | 2,89 | **prod.award** | 56 | 0,08 |
| **time.date.rel** | 2092 | 2,85 | **reference-era** | 34 | 0,05 |
| **qualifier** | 1982 | 2,7 | **loc.oro** | 31 | 0,04 |
| **org.adm** | 1793 | 2,44 | **extractor** | 29 | 0,04 |
| **pers.coll** | 1661 | 2,26 | **loc.phys.hydro** | 24 | 0,03 |
| **func.coll** | 1464 | 1,99 | **prod.doctr** | 22 | 0,03 |
| **loc.adm.nat** | 1283 | 1,75 | **century** | 22 | 0,03 |
| **loc.adm.town** | 1178 | 1,6 | **prod.serv** | 21 | 0,03 |
| **object** | 1174 | 1,6 | **prod.other** | 19 | 0,03 |
| **time.date.abs** | 1107 | 1,51 | **prod.soft** | 18 | 0,02 |
| **prod.media** | 1000 | 1,36 | **award-cat** | 11 | 0,01 |
| **demonym** | 941 | 1,28 | **loc.add.phys** | 10 | 0,01 |
| **year** | 678 | 0,92 | **zip-code** | 4 | 0,01 |
| **time.hour.rel** | 591 | 0,8 | **loc.phys.astro** | 4 | 0,01 |
| **loc.fac** | 569 | 0,77 | **prod.unk** | 2 | 0 |
| **title** | 539 | 0,73 | **other-address-component** | 2 | 0 |
| **prod.art** | 456 | 0,62 | **demonym.nickname** | 2 | 0 |
| **month** | 321 | 0,44 | **pers.other** | 1 | 0 |
| **loc.adm.reg** | 289 | 0,39 | **name.middle** | 1 | 0 |
| **range-mark** | 267 | 0,36 | **loc.unk** | 1 | 0 |

TABLE 7.12 – Nombre (NB) et proportion (%) des types d'annotations au sein d'Etape

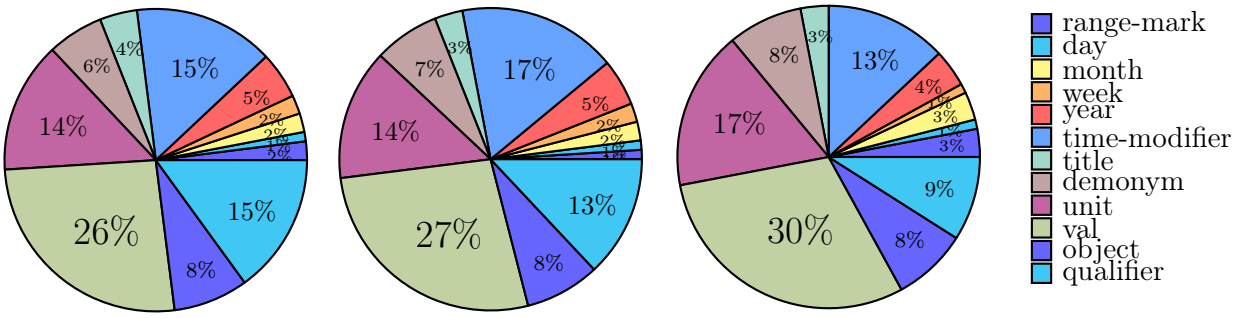
lisées pour constituer les trois parties du corpus `Etape-Train`, `Etape-Dev` et `Etape-Test`. Le nombre d'entités nommées rapporté au nombre de tokens du corpus est ici de 12,3%, dont 4,8% pour les entités nommées et 7,5% pour les composants. Nous voyons en table 7.12 le nombre et la proportion des annotations au sein de la totalité du corpus Etape (hors Quaero). Les données présentent des disparités à ce niveau de détail. Lorsque nous nous focalisons sur les types principaux d'entités nommées, la figure 7.11 nous indique qu'elles sont assez bien réparties, au travers des types comme au travers des parties du corpus. Globalement, ce corpus, quoiqu'assez volumineux, paraît bien équilibré en entités nommées.




(a) Etape-Train (b) Etape-Dev (c) Etape-Test

FIGURE 7.11 – Répartition des types d'entités nommées pour chaque partie d'Etape

L'analyse des catégories principales de composants mettent en évidence une très forte proportion pour les composants `name` (dont `name.first` et `name.last`) et `kind`, qui représentent, respectivement, 48% et 18% des composants. A l'opposé, les composants `award-cat`, `extractor`, `reference-era` et `century` pèsent moins de 1%. Nous écartons ces composants afin d'obtenir une vision de la répartition de ceux qui ne sont ni trop génériques, ni trop spécifiques, comme présenté dans le graphique 7.12.




(a) Etape-Train (b) Etape-Dev (c) Etape-Test

FIGURE 7.12 – Répartition des types de composants pour chaque partie d'Etape

Nous voyons que les proportions de composants varient de manière non négligeable selon les parties du corpus, en particulier `qualifier` et `time-modifier`. Dans l'ensemble,

étant donné que le projet Etape est précurseur dans l'annotation de composants d'entités nommés, il nous semble que ces statistiques permettent de supposer, a priori, que les annotations sont relativement homogènes.

#### 7.3.3.3 Métriques d'évaluation et performances des systèmes

Comme pour la campagne Ester2, le SER a été utilisé pour évaluer les soumissions des systèmes. Cependant l'imbrication d'annotation rend plus complexe la recherche d'appariements entre les entités nommées des soumissions et celles de l'annotation manuelle. La réalisation de l'outil d'évaluation a ainsi fait appel à des techniques de programmation dynamique [Galibert *et al.*, 2011], dont nous nous inspirons pour partie en section 9. Cette campagne d'évaluation s'achevant lors de la rédaction de ce manuscrit, nous reportons les performances des systèmes en cours d'adjudication (en date du 23 octobre 2012) [11]. Par ailleurs, si cette campagne nous donne un aperçu de la performance globale de mXS, nous nous placerons dans une configuration différent pour mener nos expérimentations en section 9.3, en écartant la partie `Etape-Test` soumise à l'adjudication.

| Part. | **Manuel** | **Rover** | **WER 23** | **WER 24** | **WER 25** | **WER 30** | **WER 35** |
|---|---|---|---|---|---|---|---|
| **1** | 85,6 | 98,1 | 100,7 | 94,2 | 98,9 | 98,4 | 100,9 |
| **2** | 156,6 | 147,4 | 178,8 | 160,4 | 168,0 | 163,9 | 168,2 |
| **3** | 36,6 | 57,2 | 59,3 | 64,7 | 62,0 | 61,7 | 71,8 |
| **4** | 50,5 | 88,0 | 98,8 | 76,8 | 92,8 | 94,9 | 99,6 |
| **5** | 44,8 | 69,7 | 73,8 | 72,1 | 73,7 | 74,8 | 86,0 |
| **6** | na | 79,2 | 79,5 | 66,8 | 80,8 | 80,0 | 87,0 |
| **7** | na | 67,8 | 68,4 | 67,6 | 70,9 | 69,9 | 85,2 |
| **8** | 37,5 | na | na | na | na | na | na |
| **9** | 62,5 | 75,8 | 79,2 | 76,9 | 79,8 | 80,5 | 90,5 |
| **10** | 39,3 | 65,0 | 69,9 | 66,3 | 70,5 | 69,9 | 87,0 |
| **CasEN** | 35,3 | na | na | 68,4 | na | na | na |
| **mXS** | 38,4 | 63,7 | 67,5 | 64,1 | 69,1 | 68,6 | 80,4 |
| **Hybrid** | 51,6 | na | na | 72,7 | na | na | na |

TABLE 7.13 – SER de la campagne Etape par type de transcription

La table 7.13 reporte les scores obtenus par les systèmes (anonymisés) qui nous ont été communiqués. Nous voyons de manière générale que les systèmes obtiennent des taux d'erreurs bien plus importants que lors de la campagne Ester2. Ceci peut être lié à la nature des transcriptions (spontanées, cette difficulté ayant également été constatée dans les performances en reconnaissance automatique de la parole) et à la nouvelle complexité des annotations à réaliser (étendues et structurées). Lorsque l'on considère les transcriptions automatiques, nous voyons que tous les systèmes présentent des dégradations assez importantes, que l'on peut raisonnablement mettre sur le compte des erreurs issues de la reconnaissance automatique de la parole.

Pour l'annotation manuelle, nous voyons que mXS se positionne quatrième, derrière CasEN, un système CRF (3) et un système qui combine CRF et PCFG (8). A 1,8 points du meilleur système sur ces données, nous considérons que mXS est compétitif. Pour les transcriptions automatiques, nous constatons que les performances de mXS se dégradent

11. De ce fait, ces résultats ne peuvent être considérés comme "officiels" : nous invitons les lecteurs intéressés à consulter les publications à paraître sur ce sujet. Nous remercions Olivier Galibert (LNE), Matthieu Carré (ELDA) et Guillaume Gravier (IRISA) pour avoir mis ces évaluations à notre disposition.

moins que d'autres systèmes, ce qui lui fait gagner des places au classement : le système passe en seconde (Rover, WER 23, WER 25, WER 30, WER 35) ou en première (WER 24) position. Il semble donc que mXS présente une bonne robustesse par rapport à d'autres systèmes. Outre le fait de mettre en œuvre une approche orientée données (réputée plus robuste), nous émettons ici l'hypothèse que la recherche séparée de début et de la fin des entités nommées (marqueurs d'annotation) permet de mieux faire face au bruit. Nous attendons l'adjudication finale pour tirer plus de conclusions de cette campagne d'évaluation et nous bornerons ici à interpréter ces résultats comme indication que l'approche est une alternative crédible pour reconnaître des entités nommées étendues et structurées dans un contexte potentiellement bruité.

### 7.3.4 Données expérimentales

La méthodologie que nous adoptons pour mener nos expériences est très similaire à celle des campagnes d'évaluation, afin de nous situer dans un cadre comparable. Cependant, notre approche explorant en profondeur les données afin d'en extraire des règles d'annotation et paramétrer le modèle numérique, nous n'avons pas recours à tous les corpus disponibles. Plus précisément, nous utilisons pour nos expériences les corpus suivants extraits des projets Ester2 et Etape :

- Ester2 :
  - Exploration : fusion d'`Ester2-Dev` et d'`Ester2-Held`
  - Test : `Ester2-Corr`
- Etape :
  - Exploration : `Etape-Train`
  - Test : `Ester2-Dev`

Sauf mentions particulières, ce seront ces données qui seront utilisées lors des expériences détaillées que nous rapportons. Pour Ester2, nous bénéficions d'un corpus de développement relativement volumineux pour explorer les données et nous évaluons sur une partie corrigée par nos soins. Comme indiqué précédemment, nous n'exploitons pas la partie `Ester2-Train`, qui est trop bruitée. Pour Etape, les expériences ayant été configurées en préparation à la campagne d'évaluation, nous ne disposions pas encore du corpus `Etape-Test` et nous évaluons donc sur `Ester2-Dev`. Par ailleurs, nous n'exploitons pas le corpus `Etape-Quaero`, trop volumineux pour notre implémentation sur les machines dont nous disposons. Pour Etape, les résultats expérimentaux que nous présenterons devront ainsi être interprétés au regard du relativement faible volume de données que nous explorons et à la lumière des performances mesurées dans le cadre de la campagne d'évaluation.

# Chapitre 8

# Extraction des règles d'annotation

## 8.1 Architecture et structure de données

### 8.1.1 Architecture générale

L'exploration de données que nous avons formalisée dans la partie précédente a conduit à l'implémentation d'un système d'extraction de règles d'annotation. Les règles obtenues à l'aide de ce module seront ensuite utilisées pour la reconnaissance d'entités nommées, que nous expérimentons sur les corpus Ester2 et Etape. Or, l'extraction de règles d'annotation est un processus qui devient rapidement consommateur en temps de calcul et potentiellement sensible au paramétrage du modèle utilisé. Nous décrivons ici la réalisation et l'implémentation effective de ce module.

L'architecture générale du système est relativement simple. Effectivement, comme indiqué en section 5.2, nous nous appuyons sur des données qui sont segmentées en énoncés et ont été automatiquement enrichies (avec disjonctions exclusives) afin de constituer notre base de données comme un multi-ensemble $\mathcal{D}$. La généralisation hiérarchique $\geq_{hi}$ est automatiquement déduite, comme indiqué en section 5.3.1, des préfixes de chaque token enrichi. La liste des marqueurs est disponible afin de les distinguer des tokens.

Pour explorer les données, le système tient compte de divers paramètres, dont les seuils de fréquence et de confiance, le mode d'exploration (mode spécifique pour l'approche par segments), l'alphabet des marqueurs, etc. Ceux-ci sont donc également fournis en paramètre au système, comme l'illustre la figure 8.1. Nous appelons notre système `mineXtract`, en référence au processus de fouille, d'exploration et d'extraction des données.

Le processus, comme nous le détaillerons en section 8.2, consiste à déterminer itérativement les motifs fréquents en augmentant graduellement leur taille. Des motifs *candidats* sont générés et testés dans les données afin de ne retenir que ceux qui sont suffisamment fréquents (en anglais, approche *generate-and-test*). Dans ce cadre, la base de données étant parcourue à chaque itération pour y relever des occurrences de motifs, la rapidité du prototype sera liée à la manière dont sont appariés les données et les motifs.

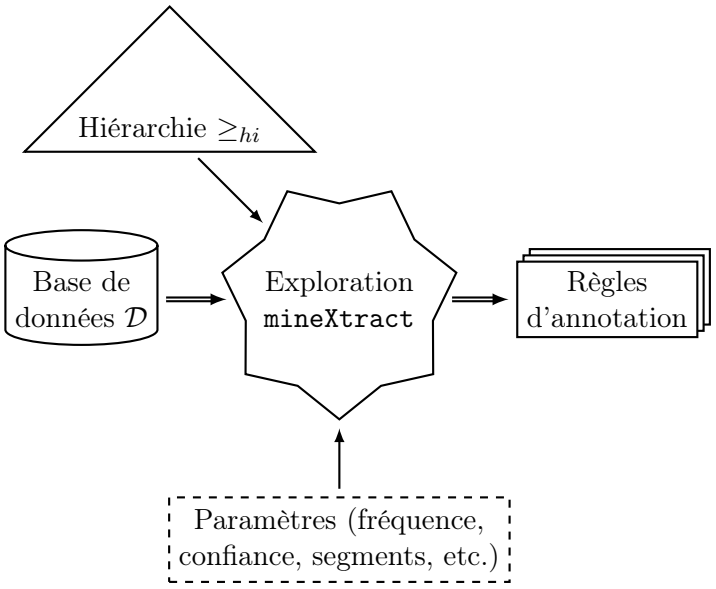



FIGURE 8.1 – Processus d'extraction de motifs

### 8.1.2 L'arbre des préfixes communs

Les motifs doivent donc être représentés selon une structure de données qui permette d'accéder rapidement à un motif particulier. Nous utilisons un arbre des préfixes communs (en anglais *trie*) qui, comme son nom l'indique, indexe les motifs selon leurs préfixes. La figure 8.2 décrit cette représentation (et fréquences associées entre parenthèses) à partir des exemples *'Il rencontra <pers> Georges Pompidou'*, *'rencontra <pers> Valéry Giscard'*, *'rencontra <date> aujourd'hui'*, *'rencontra le président'* et *'rencontra le ministre'*. Indépendamment de ces exemples, rappellons ici que, par généralisation *sur marqueurs*, les séquences sont comptabilisées avec *et* sans leurs marqueurs.

Nous voyons que cet arbre factorise les préfixes communs entre motifs. Il suffira, en fin de processus, de parcourir cet arbre en profondeur afin de récupérer les motifs extraits et leurs fréquences associées. Cet arbre est implémenté à l'aide d'une structure de données, le *nœud* de l'arbre. À un nœud `n`, sont associés les attributs suivants :

- `n.item` : donnée enrichie, élément de $\Sigma_p$ (contenu des nœuds dans la figure 8.2),
- `n.freq` : fréquence associée au motif construit depuis la racine de l'arbre (chiffres entre parenthèses dans la figure 8.2).
- `n.alt` : pointeur vers les alternatives possibles (flèches verticales dans la figure 8.2),
- `n.suiv` : pointeur vers les suffixes possibles (flèches horizontales dans la figure 8.2),

Dans cet arbre, la hiérarchie n'est pas explicitement prise en compte. Effectivement, si nous considérons deux motifs `'A B/C'` et `'A B/D'`, ainsi que leur généralisation commune `'A B'`, nous remarquons qu'il est possible que les deux motifs soient infréquents dans les données, mais que leur généralisation commune soit néanmoins fréquente. Par ailleurs, les fréquences de `'A B/C'` et `'A B/D'` ne peuvent être déduites de `'A B'` seul. La structure doit donc pouvoir systématiquement représenter et comptabiliser tous les symboles de $\Sigma_p$,

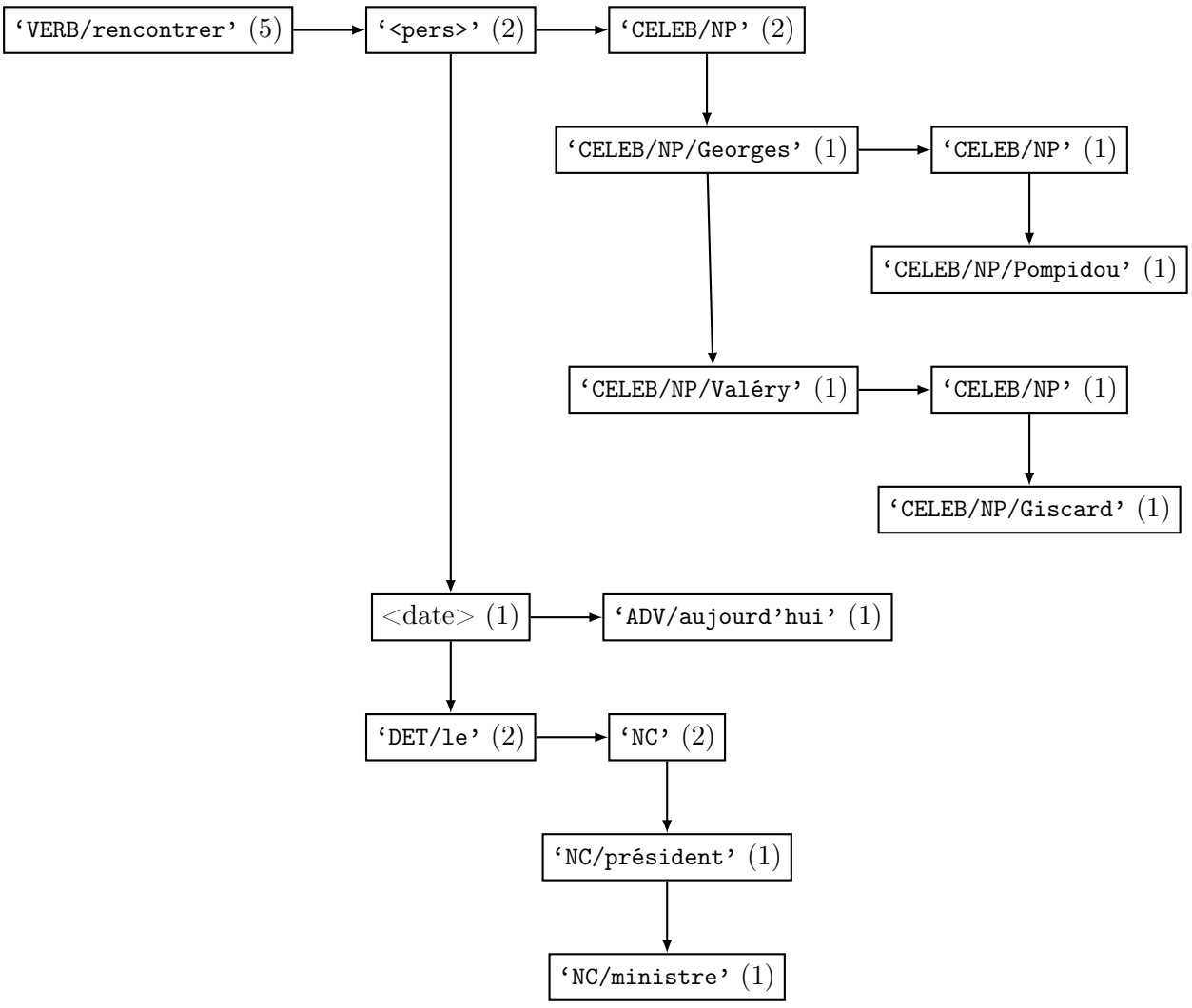



FIGURE 8.2 – Arbre des préfixes communs

quitte à opérer des optimisations lors de la construction de l'arbre.

## 8.2 Algorithmes par niveaux

### 8.2.1 Itérations sur la taille des motifs

Pour construire cet arbre des préfixes communs, de nombreuses approches en fouille de données exploitent la propriété d'*anti-monotonie* (c.f. 5.5.2) afin de procéder par *niveaux* [Agrawal et Srikant, 1995, Pei *et al.*, 2004]. En effet, cette propriété stipule qu'aucun motif $P = p_1 \dots p_n$ ne peut être plus fréquent que $p_1 \dots p_{n-1}$ ou $p_2 \dots p_n$. Ainsi, quelque soit le seuil de fréquence considéré, si un motif $P$ n'est pas fréquent, aucun motif construit par affixation de $P$ ne pourra être fréquent. Il s'ensuit que, pour déterminer l'ensemble des motifs de taille $n$, nous pouvons n'examiner que ceux dont les sous-séquences de taille $n-1$ sont fréquents, appelés motifs *candidats*.

Les *niveaux* correspondent donc à un partitionnement des motifs selon leur taille. L'exploration itère sur la taille des motifs : à partir d'un niveau $n$, tous les motifs *candidats* de taille $n+1$ sont générés, leurs fréquences sont relevées dans les données puis confrontées au seuil requis afin d'écarter ceux qui s'avèrent ne pas être suffisamment fréquents dans les données. Cette approche nous épargne de représenter et de tester tous les motifs de $(\Sigma_p)^*$, mais il sera alors crucial de pouvoir, à chaque itération, parcourir très rapidement la base de données afin d'y relever les occurrences des motifs pour un niveau donné.

### 8.2.2 Relever les occurrences dans la base de données

Les travaux de [Wang et Han, 1997] présentent plusieurs techniques afin de rechercher des occurrences de motifs au sein de séquences. Selon une méthode similaire, `mineXtract` itère sur les données (chaque énoncé et chaque token en leur sein) et réalise la recherche d'occurrences de motifs. Pour chaque token, une procédure récursive permet fait la correspondance, à la position donnée, entre l'arbre des préfixes communs et la portion de l'énoncé démarrant à cette position. Chaque motif pour lequel une occurrence est relevée verra sa fréquence incrémentée. A cet effet, nous implémentons une structure de donnée pour un token `t` en le munissant des attributs suivants :

- `t.item` : un symbole de la donnée enrichie, élément de $\Sigma_p$,
- `t.marq` : indique si `t.item` est un marqueur, élément de $\Sigma_m$,
- `t.alt` : pointeur vers la liste chaînée des prochains items de la disjonction exclusive,
- `t.suiv` : pointeur vers la liste chaînée des prochains tokens de l'énoncé,

L'algorithme 1 présente une version simplifiée de la procédure $tokenMotifs$ qui relève les occurrences de motifs à une position donnée d'un énoncé, aux marqueurs près. Remarquons que, pour optimiser ces parcours, l'algorithme suppose que certaines données ont préalablement été triées : les généralisations hiérarchiques retournées par la fonction *parents* et les alternatives au sein de l'arbre des préfixes communs. De surcroît, il nous faut tenir compte de deux particularités afin que l'anti-monotonie soit respectée :

- **Séquences de marqueurs** : la généralisation *aux marqueurs près* suppose d'*énumérer* les sous-séquences de marqueurs, ce que nous réalisons par passage du paramètre $I$.

– **Disjonction exclusive** : la procédure *altParents* permet de vérifier qu'il n'y a pas de comptage de doublons au sein d'une disjonction exclusive (lorsque deux items ont une généralisation commune).

**Algorithme 1:** Recherche des occurrences de motifs dans les données ($tokenMotifs$)

```
Données : token t, nœud n, longueur l, items I
si t.alt alors                                              /* ou exclusif */
    tokenMotifs(t.alt, n, l, I)
fin
si t.marq alors                              /* généralisation sur marqueurs */
    tokenMotifs(t.suiv, n, l, I ∪ t.item)
fin
si t.item ∉ I alors
    p ← parents(t.item)
    tant que p et p.item ∉ altParents(t) faire  /* généralis. hiérarchique */
        tant que n et p.item > n.item faire          /* recherche du motifs */
            n ← n.alt
        fin
        si n et p.item = n.item alors
            si l > 1 alors
                tokenMotifs(t.suiv, n.suiv, l − 1, ∅)
            fin
            sinon
            fin
            n.freq = n.freq + 1
        fin
        p ← p.parent
    fin
fin
```

Cet algorithme est au cœur de `mineXtract`. Nous en implémentons également une version adaptée à l'extraction de motifs de segments, qui associe à un item de motif autant de tokens que possible dans les données. Dans tous les cas, l'algorithme est précédé d'une procédure de génération des candidats et suivi d'une procédure d'élagage de l'arbre des préfixes. Ainsi, nous sommes en mesure de comptabiliser, par niveaux, les occurrences des motifs dans les données, jusqu'à ce qu'il n'y ait plus de motifs fréquents. Au terme de l'exploration, il est trivial de parcourir l'arbre des préfixes communs afin d'extraire les règles d'annotation informatives selon les critères présentés en section 5.5.4.

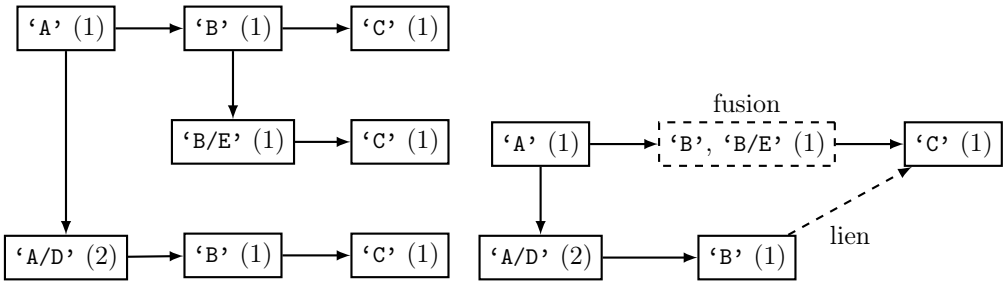
### 8.2.3 Implémentation, exécution et optimisations

Nous implémentons `mineXtract` en C++ et le testons sur les corpus dont nous disposons. Pour ce faire, les machines que nous utilisons disposent généralement de plusieurs cœurs cadencés à 2-3GHz (mais l'implémentation n'est pas distribuée) et une mémoire

vive de 30-40Go. Le corpus est enrichi selon les traitements morpho-syntaxiques et lexicaux décrits en 7.2.1 et en 7.2.2. L'approche que nous adoptons nous contraint à explorer exhaustivement tous les motifs fréquents (et non seulement les règles d'annotation). Ainsi, au travers des itérations, seul le critère de fréquence est pris en compte pour filtrer les motifs. En fin de processus, la confiance est calculée afin d'extraire les règles fréquentes et confiantes.

L'arbre des préfixes est donc construit au fur et à mesure de l'exploration. Nous pouvons en mesurer la taille à chaque niveau selon le nombre de nœuds correspondants à des motifs candidats générés ou des motifs fréquents retenus. Nous nous apercevons cependant qu'il est possible d'optimiser la structure de l'arbre afin d'en limiter la taille et le temps de parcours [Wang et Han, 2004, Qian *et al.*, 2010, Bonchi et Lucchese, 2005]. Effectivement, un nœud de l'arbre peut stocker plusieurs motifs lorsque ceux-ci ont exactement les mêmes occurrences dans les données. Or, pour deux motifs, nous pouvons affirmer qu'ils ont mêmes occurrences dans les données s'ils sont généralisation l'un de l'autre et qu'ils ont même fréquence. Selon ce principe, deux optimisations se sont avérées simples à implémenter et efficaces :

- **Fusion de suffixes** : deux nœuds feuilles de l'arbre d'un même parent qui ont même fréquence et sont généralisation l'un de l'autre peuvent être fusionnées.
- **Lien de préfixes** : deux nœuds de parents distincts qui ont même fréquence et dont les motifs sont généralisation l'un de l'autre auront les mêmes nœuds enfants.

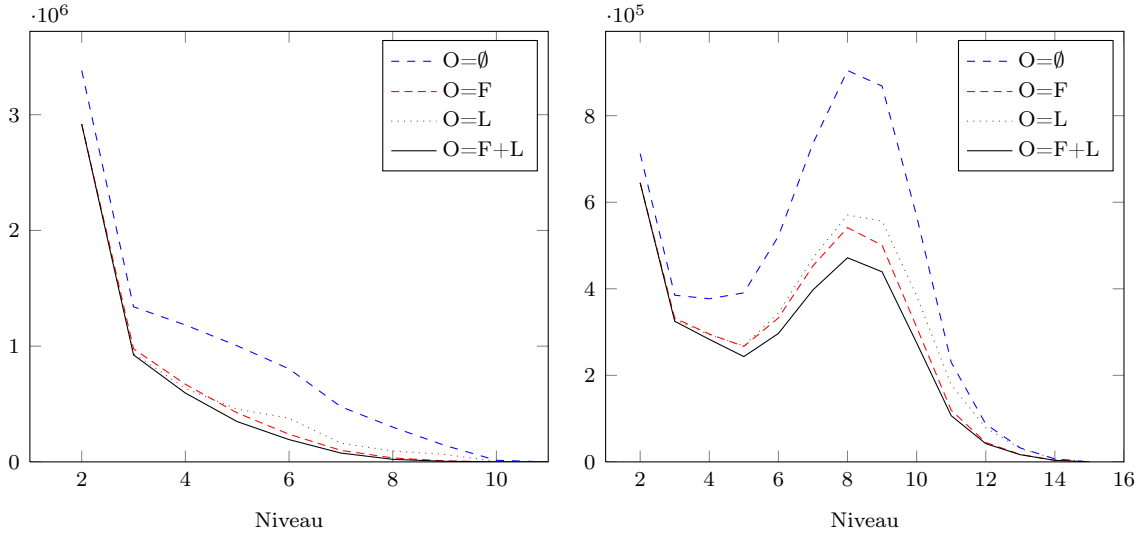



(a) Avant optimisations (b) Après optimisations

FIGURE 8.3 – Optimisations de l'abre des préfixes

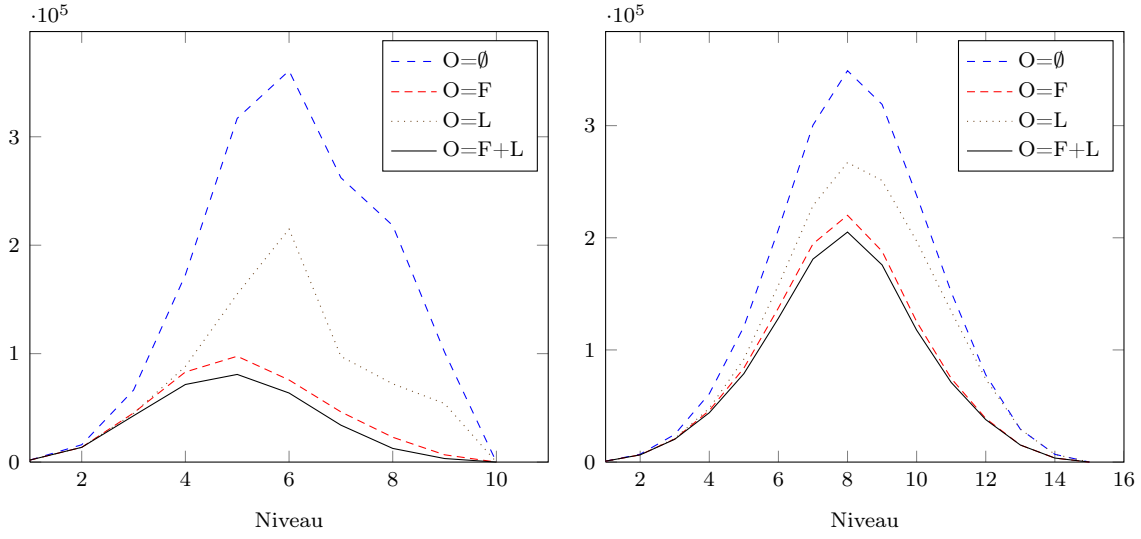
La figure 8.3 illustre ces deux techniques sur un exemple minimal. Les nœuds feuilles des motifs `'A B'` et `'A B/E'` peuvent être fusionnés. De la même manière, les nœuds feuilles des motifs `'A B'` et `'A/D B'` peuvent être liés au même nœud enfant `'C'`. Nous réalisons ces opérations après la phase de recherche des occurrences, simultanément à l'élagage des nœuds non fréquents de l'arbre.

Nous mesurons dans la figure 8.4 les gains réalisés en terme de nombre de nœuds candidats et fréquents de l'arbre au cours des itérations. La fusion de suffixes semble efficace, en particulier pour limiter le nombre de nœuds fréquents, c'est à dire les nœuds qui sont à la fois conservés en mémoire et utilisés pour générer des candidats. Les liens préfixes paraissent plus efficaces pour la génération de candidats. Ces deux optimisations se recouvrent (une fusion non réalisée à un niveau donné correspondant à un lien à un niveau ultérieur) et nous voyons que, de concert, nous obtenons une réduction du nombre de nœuds conséquente à




(a) Nœuds candidats sur Ester2, $F \geq 14$

(b) Nœuds candidats sur Etape, $F \geq 83$




(c) Nœuds candidats sur Ester2, $F \geq 14$

(d) Nœuds candidats sur Etape, $F \geq 83$

FIGURE 8.4 – Nombre de noeuds par niveaux selon les Optimisations (O) de fusion (F) et de liens (L)

chaque itération. Sans entrer plus dans les détails, les expériences que nous avons conduites montrent que ces optimisations sont intéressantes à mettre en œuvre dans la perspective d'explorer les données à basse fréquence, lorsque la combinatoire de la hiérarchie produit de nombreuses redondances au sein des motifs les plus longs.

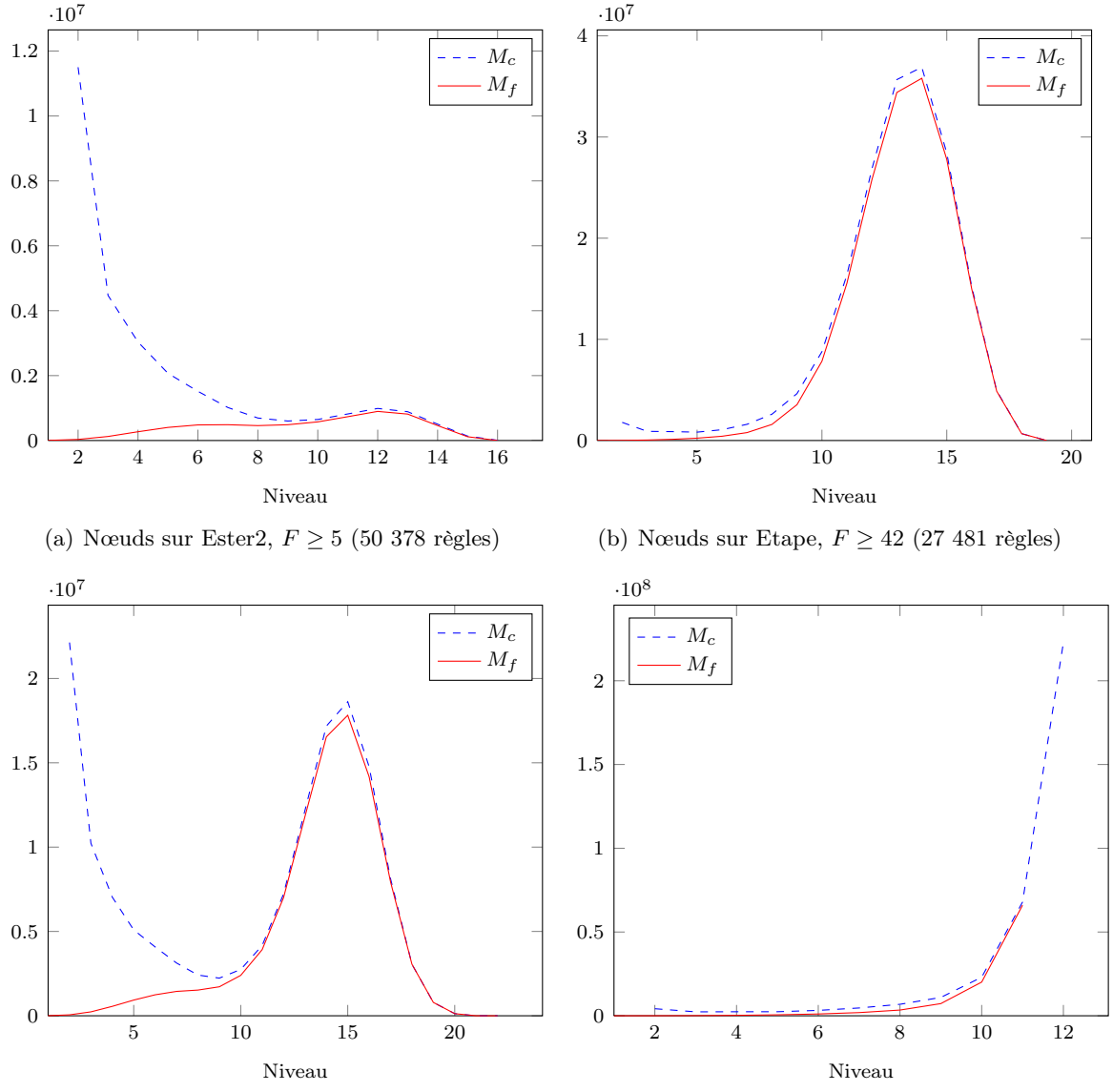
## 8.3 Etude des règles d'annotation extraites

Nous cherchons à déterminer quels seuils de fréquence nous pouvons atteindre. La figure 8.5 identifie les limites de l'exploration de ce point de vue sur les corpus Ester2 et Etape. Les graphiques (a), (b) et (c) montrent le comportement lorsque l'extraction atteint son terme. Nous constatons que le nombre de nœuds candidats, s'il diminue lors des premières itérations, n'est pas strictement décroissant et connaît un accroissement en cours d'exploration, en particulier pour Etape où il atteint plusieurs dizaines de millions entre la dixième et la seizième itération. Nous y mentionnons également, pour information, le nombre de règles d'annotation extraites en fin de processus avec un seuil de confiance de 5% et constatons à ces fréquences une forte disproportion entre le nombre de nœuds générés et le nombre de règles d'annotation finalement extraites. Dans le graphique (d), nous voyons que lorsque le seuil de fréquence est encore abaissé, l'exploration ne parvient pas à son terme : trop de nœuds sont générés, l'arbre des préfixes ne tient plus en mémoire vive, les mécanismes de pagination rendent l'exploration excessivement lente.

Nous faisons la même étude, avec les mêmes paramètres, en utilisant les motifs de segments, dont les résultats sont reportés en figure 8.6. En comparant les graphiques (a) et (b), nous remarquons en premier lieu que ces motifs nécessitent un moins grand nombre d'itérations et qu'un plus grand nombre de règles d'annotation sont extraites. Dans l'ensemble, la courbe indiquant le nombre de nœuds générés paraît similaire. Cependant, le comportement est différent selon le corpus : pour Ester2, plus de nœuds sont nécessaires à l'exploration, tandis que pour Etape (bien plus volumineux en tokens et en marqueurs), le nombre de nœuds requis est moins important et la recrudescence du nombre de nœuds en cours de processus est beaucoup moins accentuée. En particulier, en (d), l'exploration peut être menée à son terme. Cette formulation des motifs, avec sa prise en compte particulière de segments d'items, réduit significativement la combinatoire et est plus productive en nombre de règles extraites.

Afin de confirmer l'impact de l'utilisation de motifs de segments lors de l'exploration des données, nous réalisons la même étude lorsque le corpus est de surcroît enrichi par une analyse syntaxique de surface (réalisée par le *chunker* de TreeTagger). Les syntagmes adjectivaux, nominaux, prépositionnels et verbaux sont identifiés et catégorisés, ce qui vient former un niveau supplémentaire de la hiérarchie, au dessus de la catégorisation sémantique. Un syntagme concerne alors fréquemment plusieurs tokens, cet enrichissement supplémentaire vient ainsi accroître la combinatoire lors de l'exploration des motifs. La figure 8.7 montre alors de manière très nette la réduction du nombre de nœuds et de règles produites lors de l'exploration.

Enfin, nous nous penchons sur les règles d'annotation effectivement extraites lorsque le processus parvient à son terme, c'est à dire avec les paramètres correspondant aux graphiques (a) et (b) de la figure 8.5. La figure 8.8 présente le nombre de règles d'annotation



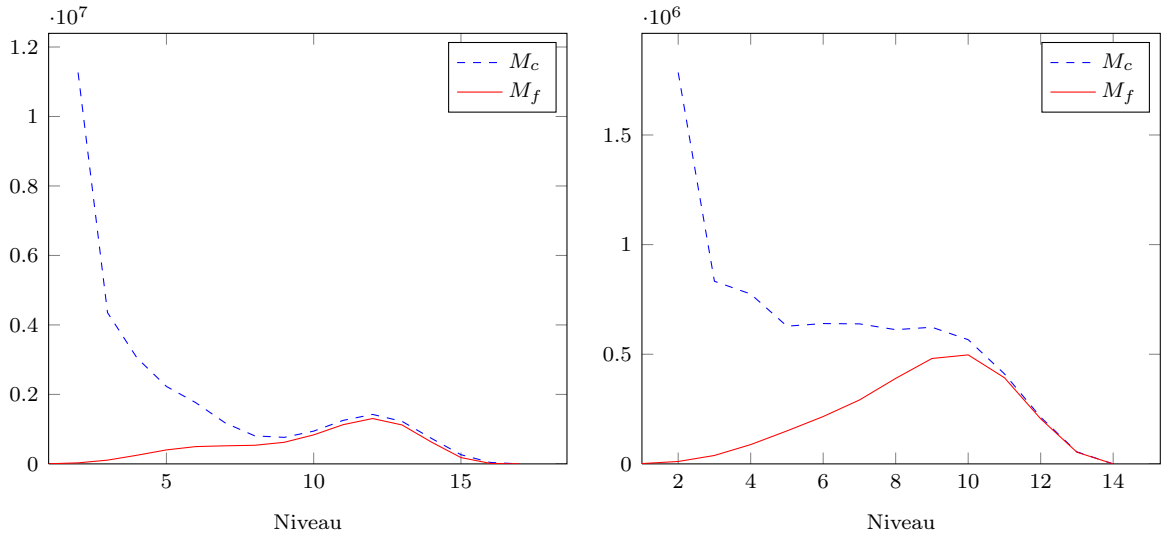

(a) Nœuds sur Ester2, $F \geq 5$ (50 378 règles)

(b) Nœuds sur Etape, $F \geq 42$ (27 481 règles)

(c) Nœuds sur Ester2, $F \geq 3$ (87 321)

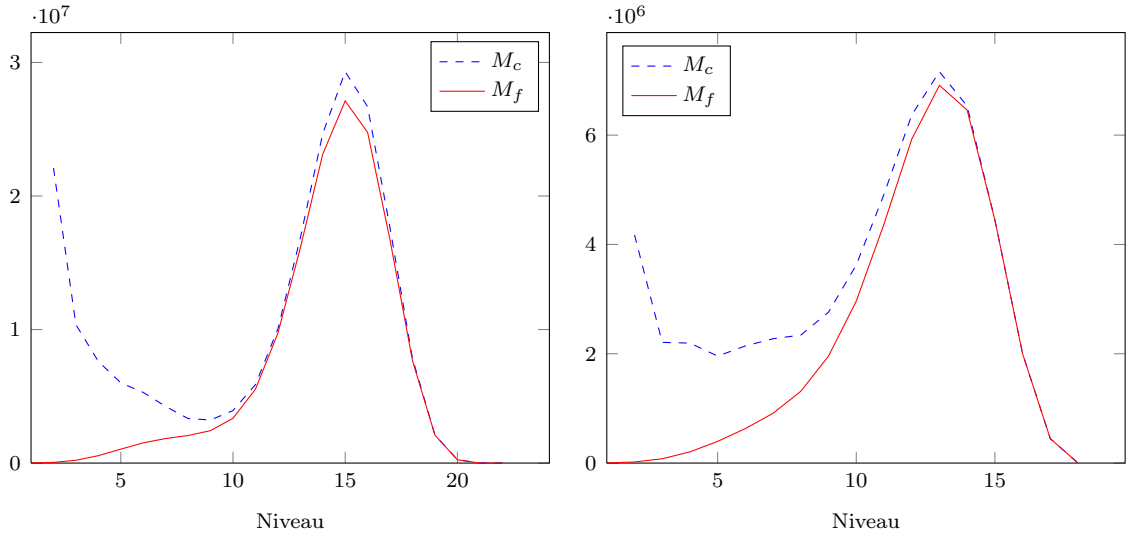
(d) Nœuds sur Etape, $F \geq 21$

FIGURE 8.5 – Nombre de nœuds candidats ($M_c$) et fréquents ($M_f$) par niveaux selon la Fréquence (F)




(a) Nœuds sur Ester2, $F \geq 5$ (103 556 règles)

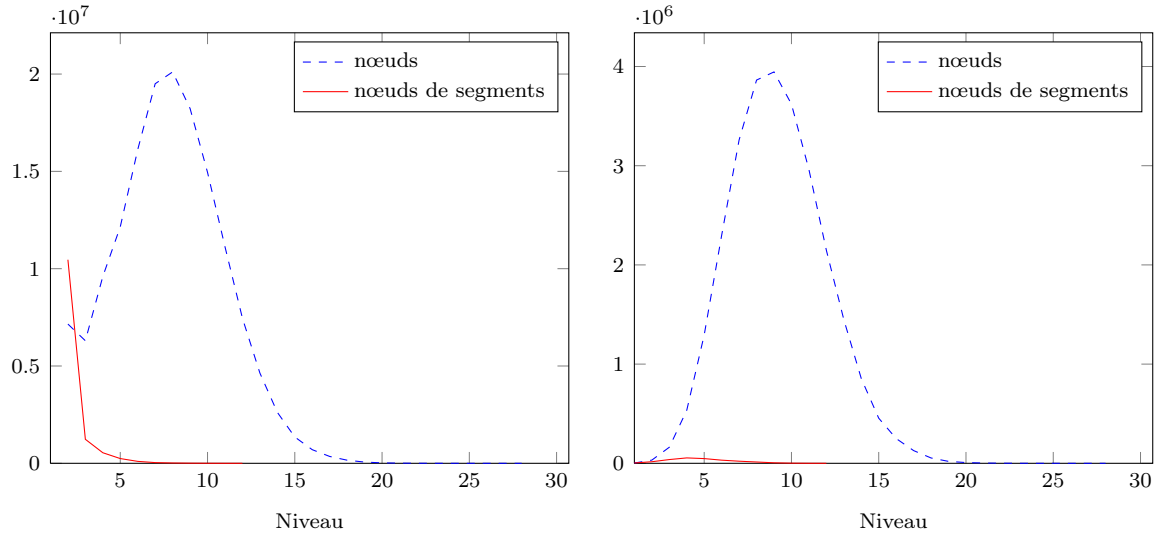
(b) Nœuds sur Etape, $F \geq 42$ (32 005 règles)




(c) Nœuds sur Ester2, $F \geq 3$ (248 024 règles)

(d) Nœuds sur Etape, $F \geq 21$ (73 176 règles)

FIGURE 8.6 – Nombre de nœuds de segments candidats ($M_c$) et fréquents ($M_f$) par niveaux selon la Fréquence (F)




(a) Candidats (b) Fréquents

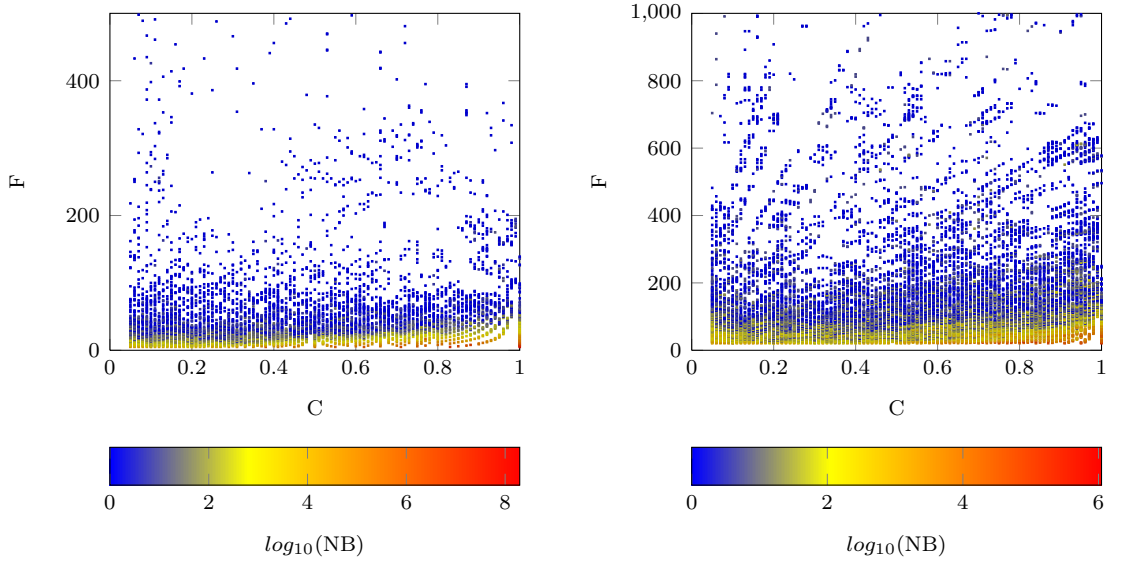
FIGURE 8.7 – Nombre de nœuds (849 385 règles) ou de nœuds de segments (15 103 règles) sur Ester2 à fréquence $F \geq 12$ et confiance $C \geq 0,1$ après enrichissment syntaxique

extraites, en densité sur une échelle logarithmique pour (a) et (b), en répartition selon la fréquence pour (c) et (d), en répartition selon la confiance pour (e) et (f). Sans surprises, le nombre de règles extraites croît fortement à mesure que la fréquence est abaissée. En revanche, ce nombre semble être également réparti selon la confiance, ce qui est plus surprenant.

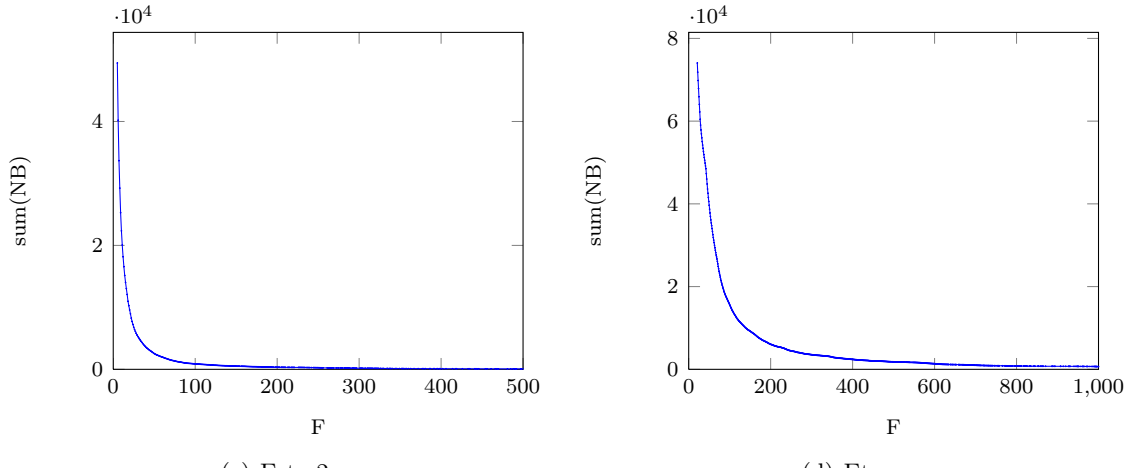
Ainsi, nous voyons qu'il sera difficile pour `mineXtract` d'explorer exhaustivement les règles avec un seuil de fréquence bas. En revanche, il est possible d'utiliser un large spectre de confiance, ce qui permettra de bénéficier de nombreux indices, même lorsqu'ils sont faiblement corrélés aux marqueurs d'entités nommées. L'influence de ces paramètres sur les performances du système sera discuté plus en détail en section 9. Pour le moment, notre problématique est d'extraire autant de règles que possible. Pour abaisser les seuils (en particulier pour Etape), nous sommes amenés à mettre en place deux contraintes supplémentaires afin de réduire la combinatoire :

- **Nombre de marqueurs** : un motif ne peut contenir plus d'un marqueur.
- **Niveaux** : le nombre d'itération de l'algorithme par niveaux est limité à 7.

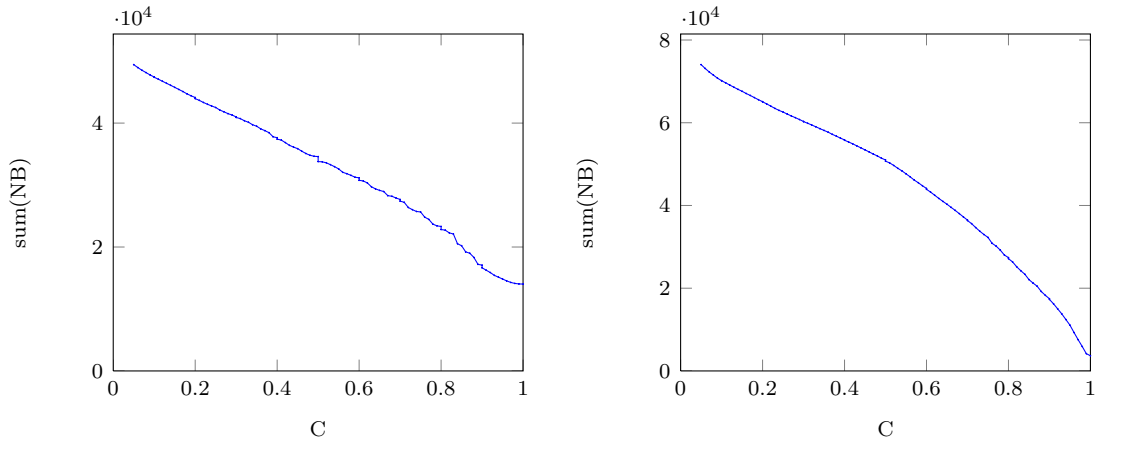
Pour l'exploration des données, nous fixons le seuil minimal de fréquence relative à $3.10^{-5}$ (en fréquence absolue 5 pour Ester2 et 13 pour Etape) et le seuil minimal de confiance est fixé dans les deux cas à 5%. Avec ces paramètres, la figure 8.9 donne des informations sur la forme que prennent les motifs selon les axes *structurels* (taille) et *ontologiques* (profondeur) évoqués en section 1.1. La longueur des règles d'annotation (marqueurs non compris) varie autour de quatre éléments, les règles les plus longues étant détectées à basse fréquence. De la même manière, la profondeur des items (somme sur les items des spécialisations au delà de la racine de la hiérarchie) se situe autour de quatre, mais est plus étalée pour atteindre des valeurs plus importantes. Dans l'ensemble, ces statistiques




(a) Ester2 (b) Etape

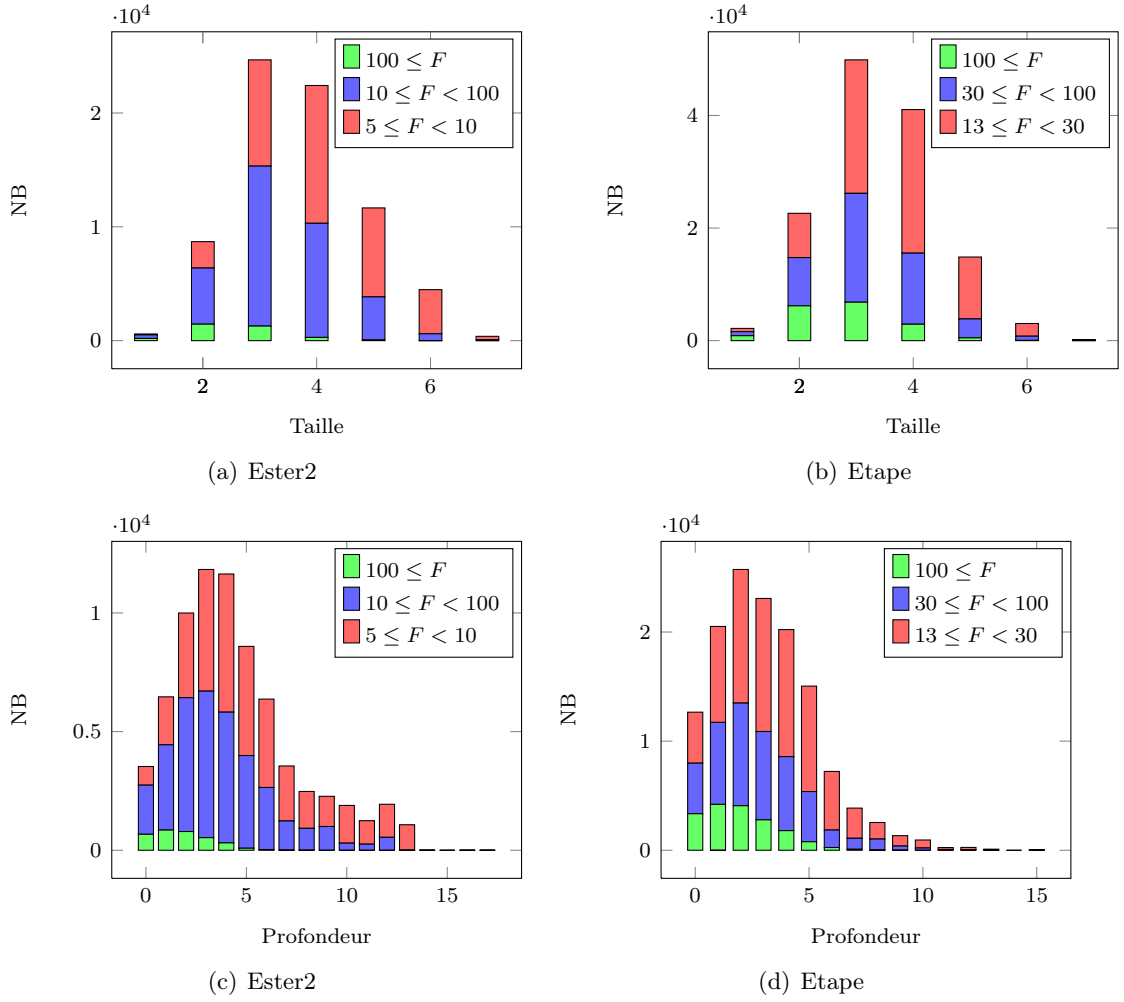



(c) Ester2 (d) Etape




(e) Ester2 (f) Etape

FIGURE 8.8 – Nombre de règles d'annotation (NB) selon la fréquence (F) et la confiance (C)




(a) Ester2 (b) Etape

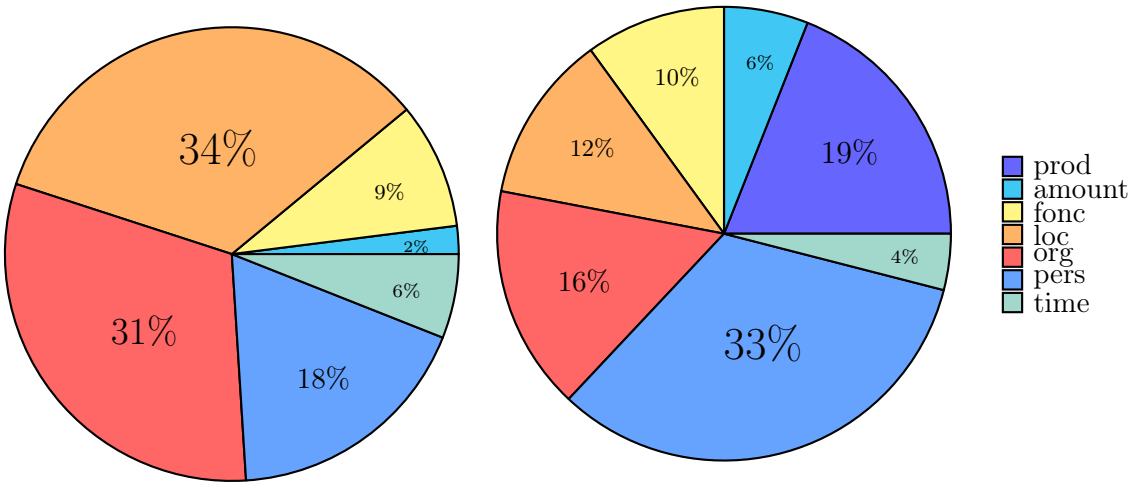
(c) Ester2 (d) Etape

FIGURE 8.9 – Nombre de règles d'annotation (NB) selon leur taille, leur profondeur et la Fréquence (F)

confirment que les règles d'annotation sont explorées sur les deux axes que nous avons définis. Les proportions sont similaires pour les règles de segments.

Nous examinons en figure 8.10 la répartition du nombre de règles d'annotation selon les types d'entités nommées. De manière générale, les proportions sont corrélées aux nombres d'entités nommées dans les corpus concernés (c.f. 7.3). Nous constatons cependant une assez nette sous-représentation des types `time` et `amount`. Il semble qu'il y ait objectivement, avec l'approche que nous mettons en œuvre, moins de descripteurs pour ces types. Nous en déduisons qu'ils sont relativement homogènes dans les données. Le type `prod`, inversement, est sur-représenté au sein des motifs extraits d'Etape (il est quasiment absent d'Ester2) et nous faisons l'hypothèse que ce type est assez hétérogène.

Avec ces paramètres, nous extrayons 59 610 règles d'annotation sur le corpus Ester2, 109 390 sur le corpus Etape. Pour les règles de segments, nous en obtenons respectivement 132 205 et 143 205. Plus généralement, nous considérons que `mineXtract` nous permet




(a) Ester2 (b) Etape

FIGURE 8.10 – Répartition des types d'entités nommées au sein des motifs

d'extraire des règles d'annotations d'intérêt pour reconnaître les entités nommées de manière efficace. Pour vérifier leur pertinence et mieux déterminer l'impact de la fréquence et de la confiance sur les performances du système `mXS`, il s'agit maintenant d'utiliser ces règles pour paramétrer un modèle de reconnaissance automatique des entités nommées.

# Chapitre 9

# Utilisation des règles pour annoter

## 9.1 Programmation dynamique

Avec les règles d'annotation que nous avons extraites, nous pouvons nous mettre dans la perspective d'annoter des textes à l'aide de ces règles selon les approches décrites en section 6. Pour ce faire, outre les règles d'annotation extraites, nous disposons des statistiques des corpus pour l'inférence bayésienne et de l'outil *scikit-learn*[1] [Pedregosa *et al.*, 2011] pour la régression logistique. Nous sommes donc en mesure d'utiliser les règles et, au besoin, d'estimer des probabilités, afin d'insérer des marqueurs et des séquences de marqueurs au sein des énoncés.

Nous considérons alors l'annotation comme un processus séquentiel qui dispose d'un *état d'annotation* au long de la séquence examinée. Ceci est illustré en table 9.1, à partir de l'exemple *'Et Pierre ira jeudi 18 novembre avec lui au Centre Georges Pompidou.'*, que l'on annote au format Etape de la manière suivante :

`'Et <pers> <name.first> Pierre </name.first> </pers> ira <time.date.abs> <week> jeudi </week> <day> 18 </day> <month> novembre </month> </time.date.abs> avec lui au <loc.fac> <name> Centre <pers.ind> <name.first> Georges </name.first> <name.last> Pompidou </name.last> </pers.ind> </name> </loc.fac>.'`.

Rechercher les solutions d'annotation à l'aide des probabilités des marqueurs repose sur deux éléments. En premier lieu, nous tenons compte du guide d'annotation (notamment les imbrications) afin déterminer les états acceptables au fur et à mesure de l'annotation. Pour Ester2, nous réduisons la complexité en interdisant toute imbrication : il n'y aura que 8 états possibles. Pour Etape, nous contraignons les états accessibles (entités et composants mélangés) selon les directives du guide d'annotation. D'autre part, l'hypothèse d'indépendance que nous faisons en section 6.4 nous permet de formaliser l'annotation comme un processus markovien : pour un token donné, la vraisemblance des états d'annotation ne dépend que des vraisemblances des états précédents, combinées avec les probabilités de transition.

Ainsi, nous sommes en mesure de déterminer l'annotation la plus vraisemblable par programmation dynamique. Nous implémentons ceci comme le module `mStruct`, qui réa-

1. `http://scikit-learn.org`

| Token | Marqueurs | État d'annotation |
|---|---|---|
| Et | ∅ | / |
| Pierre | ‘<pers.ind><name.first>’ | /pers.ind/name.first |
| ira | ‘</name.first></pers.ind>’ | / |
| jeudi | ‘<time.date.abs><week>’ | /time.date.abs/week |
| 18 | ‘</week><day>’ | /time.date.abs/day |
| novembre | ‘</day><month>’ | /time.date.abs/month |
| avec | ‘</month></time.date.abs>’ | / |
| lui | ∅ | / |
| au | ∅ | / |
| Centre | ‘</loc.fac><name>’ | /loc.fac/name |
| Georges | ‘<pers.ind><name.first>’ | /loc.fac/name/pers.ind/name.first |
| Pompidou | ‘</name.first><name.last>’ | /loc.fac/name/pers.ind/name.last |
| . | ‘</name.last></pers.ind></name></loc.fac>’ | / |

TABLE 9.1 – Marqueurs et états d'une annotation

lise l'annotation structurée à partir d'indices locaux. L'algorithme 2 en donne une version simplifiée. Une structure de données représente une hypothèse d'annotation par son état d'annotation, les marqueurs qui ont été insérés et sa probabilité. Le modèle *modele* fournit les probabilités des séquences de marqueurs à toutes positions d'un énoncé (selon les règles déclenchées). Notons que, parmi les probabilités de marqueurs se trouve ∅, la probabilité de n'insérer aucun marqueur. Les contraintes du guide d'annotation sont modélisées par une procédure *guide* qui vérifie la possibilité d'insérer des marqueurs par rapport à une hypothèse d'annotation existante. Enfin, la procédure *mettreAJourEtat* actualise l'état d'une hypothèse selon les marqueurs insérés, la procédure *ajouterOuRemplacerEtat* indexe les hypothèses en mémoire (pour ne conserver que la plus vraisemblable par état d'annotation). L'algorithme renvoie en fin d'énoncé l'hypothèse pour l'état /, c'est à dire celle pour laquelle toutes les annotations sont terminées.

**Algorithme 2:** Recherche des annotations vraisemblables par programmation dynamique (*annotationSequence*)

**Données** : Énoncé $E$, modèle d'annotation *modele*, guide d'annotation *guide*
$H \leftarrow \{(marqueurs : \emptyset, proba : 1, etat : /)\}$
**pour chaque** $0 \leq i < |E|$ **faire** `/* positions dans la séquence */`
  $H' \leftarrow \emptyset$
  **pour chaque** $(m, p) \in modele(S, i)$ **faire** `/* marqueurs et probabilités */`
    **pour chaque** $h \in H$ ***tq*** $guide(h, m)$ **faire** `/* consultation du guide */`
      $h' \leftarrow (marqueurs : h.marqueurs + m, proba : h.proba * p)$
      $h'.mettreAJourEtat()$
      **si** $\nexists h'' \in H$ ***tq*** $h''.etat = h'.etat$ ***et*** $h''.proba > h'.proba$ **alors**
        $ajouterOuRemplacerEtat(H', h')$
      **fin**
    **fin**
  **fin**
  $H \leftarrow H'$
**fin**
**retourner** $h \in H$ ***tq*** $h.etat = /$

Nous retrouvons dans le principe même de cet algorithme de `mStruct` le mécanisme des instructions liées aux marqueurs (c.f. 3.3). Ici apparaît de surcroît la notion d'*état*, qui matérialise la dynamique de l'annotation à mesure qu'un énoncé est analysé pour y reconnaître des entités nommées. Malgré la réduction du nombre de solutions examinées par utilisation de la programmation dynamique, le nombre d'hypothèse émises peut être très important, en particulier lorsque l'on autorise le modèle à fournir des probabilités très faibles pour les marqueurs les moins vraisemblables et que les imbrications sont autorisées, comme c'est le cas pour Etape. A cet effet, nous imposons à `mStruct` un seuil minimal de probabilité de 0.005 en sortie du modèle qui indique les probabilités des séquences de marqueurs et ne conservons après chaque itération que les 20 hypothèses d'annotation les plus probables.

## 9.2 Ester2

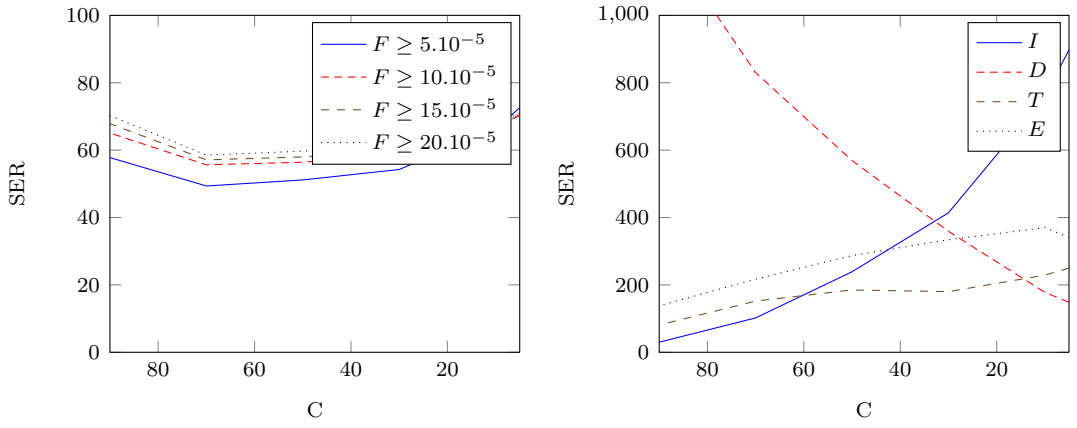
### 9.2.1 Règles confiantes

Nous réalisons des expériences par seuils de fréquence et de confiance sur le corpus Ester2. Nous savons, pour une fréquence donnée, que lorsque nous diminuons le seuil de confiance, les règles seront plus nombreuses et, a priori, d'une moins bonne précision individuellement. Faire varier la confiance nous permet ainsi d'évaluer le comportement du système face à des informations partielles. Notons que le détail des résultats expérimentaux à divers seuils de fréquence et de confiance est fourni en annexe A. Nous commençons par mener des expériences dans lesquelles toutes les règles d'annotation qui peuvent être appliquées le sont, par ordre de confiance, comme mentionné en section 6.1. Nous nommons cette approche `Règles`. A cet effet, toutes les règles d'annotations sont extraites, sans contraintes, mais seules les règles qui ne sont pas partielles (c.f. 6.1) peuvent être utilisées, ce qui en réduit le nombre à 18 568 aux plus bas seuils en fréquence et confiance. Nous notons ici que cette contrainte supplémentaire sur les règles pourrait avantageusement être mise en place lors de l'extraction, ce qui résulterait en un moindre filtrage.

Les graphiques de la figure 9.1 indiquent qu'abaisser le seuil de fréquence permet d'augmenter les performances. Nous pouvons donc explorer les données de manière à obtenir, à même confiance, autant de règles d'annotation que possible. En revanche, cette approche tolère difficilement les règles peu confiantes : la performance globale se dégrade dès que l'on utilise des règles en dessous de 70% de confiance. Dans le graphique des erreurs, nous voyons clairement la difficulté : dès lors que des règles moins confiantes sont utilisées, de nouvelles entités nommées sont reconnues (le nombre d'erreurs de délétions est diminué),mais nous nous apercevons que nombre d'entre elles correspondent à du bruit (nombre croissant d'erreurs d'insertion).

### 9.2.2 Inférence bayésienne

Il nous faut alors être en mesure de mieux tenir compte des règles d'annotation qui sont peu confiantes, mais qui peuvent néanmoins participer à la reconnaissance d'entités nommées. Nous mettons donc en place le modèle probabiliste (c.f. 6.3) afin d'estimer les




(a) Performances

(b) Erreurs, $F \geq 5.10^{-5}$

FIGURE 9.1 – Performances (SER) erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E) selon la Fréquence relative (F) et la Confiance (C) avec l'approche `Règles` sur Ester2

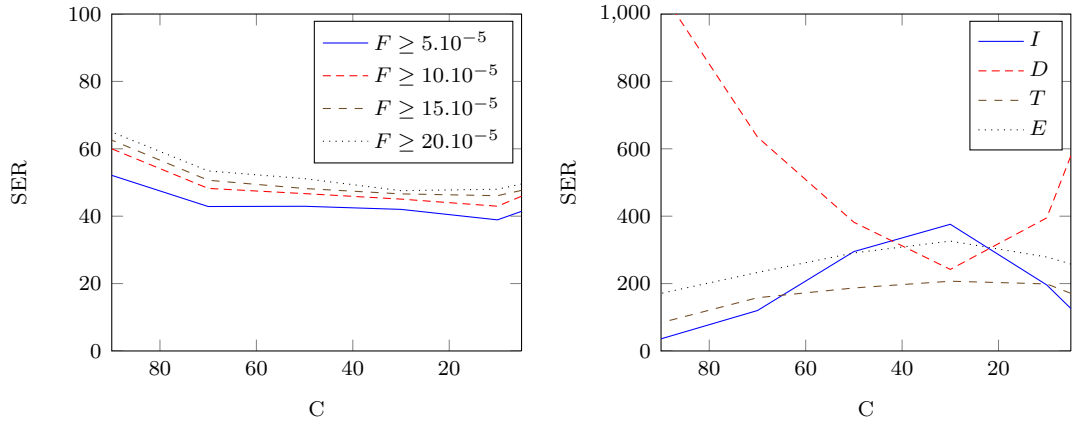
probabilités de réaliser des transductions. A cet effet, nous implémentons l'inférence bayésienne (appelée `Bayes`), qui présente l'avantage de pouvoir être entièrement calculée à partir des statistiques dont nous disposons en fin d'extraction (il n'est pas nécessaire d'ajuster les paramètres itérativement). Nous pouvons maintenant n'utiliser que des règles partielles et ajoutons lors de l'extraction les contraintes nous permettant de réduire le nombre de règles extraites (c.f. 8.3).

Dans la figure 9.2, nous voyons que cette approche nous permet d'obtenir de meilleures performance globales, par une meilleur prise en compte de la moindre confiance associée à certaines règles lors de l'annotation. Effectivement, nous constatons que les courbes concernant les erreurs d'insertion et de délétion s'inversent lorsque la confiance diminue. Si ce modèle donne de meilleurs résultats que le précédent, il semble cependant difficile de déterminer l'influence a priori des règles locales et d'équilibrer correctement les divers types d'erreurs commises.

### 9.2.3 Régression logistique

Enfin, nous mettons en place le modèle, appelé `Logit`, qui estime les probabilités des marqueurs par ajustement de poids à l'aide de la régression logistique. Ainsi, nous n'exploitons plus les statistiques collectées lors de l'extraction des règles d'annotation, mais sommes contraints de réaliser l'ajustement des poids du modèle séparément. Cet ajustement est conduit sur le même corpus, à l'aide de l'outil *scikit-learn*, avec l'objectif d'obtenir un modèle qui estime mieux les probabilités des marqueurs à l'aide des nombreuses règles d'annotation dont nous disposons. Lorsque nous utilisons les motifs de segments, cet approche est appelée `Logit+Segs`.

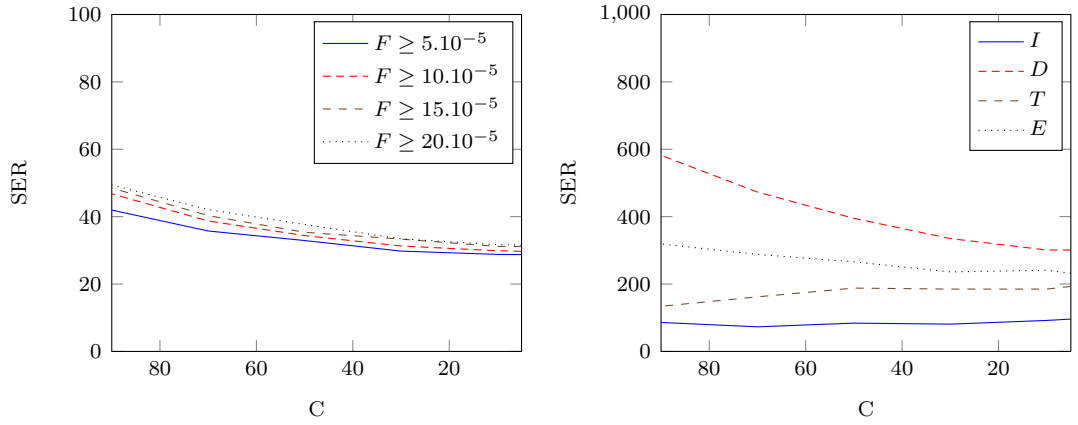
Les figures 9.3 et 9.4 présentent les résultats que nous obtenons avec cette approche. Les performances sont encore améliorées et nous remarquons que les diverses erreurs évoluent




(a) Performances

(b) Erreurs,$F \geq 5.10^{-5}$

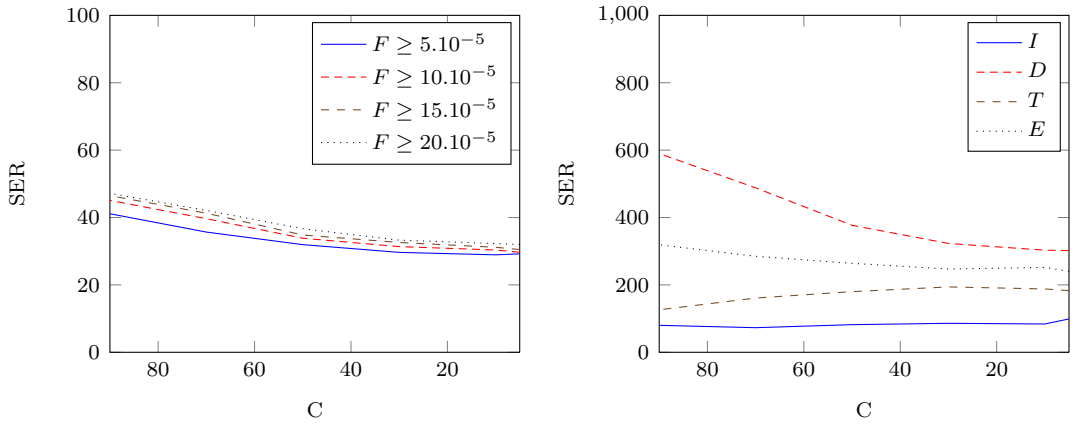
FIGURE 9.2 – Performances (SER) erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E) selon la Fréquence relative (F) et la Confiance (C) avec l'approche `Bayes` sur Ester2




(a) Performances

(b) Erreurs, $F \geq 5.10^{-5}$

FIGURE 9.3 – Performances (SER) erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E) selon la Fréquence relative (F) et la Confiance (C) avec l'approche `Logit` sur Ester2




(a) Performances

(b) Erreurs, $F \geq 5.10^{-5}$

FIGURE 9.4 – Performances (SER) erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E) selon la Fréquence relative (F) et la Confiance (C) avec l'approche avec motifs de segments `Logit+Segs` sur Ester2

de manière plus stable lorsque le seuil de confiance est modifié. Par ailleurs, à mesure que ce seuil est abaissé, nous constatons que `mXS` commet essentiellement moins d'erreurs de délétion, ce qui nous permet d'affirmer qu'il pondère efficacement les divers indices à sa disposition afin de réduire graduellement le silence. Sur ce corpus, nous ne constatons pas de grande différence par utilisation de segments, malgré le fait qu'ils produisent un plus grand nombre de règles d'annotation.

En regardant le détail des erreurs commises par type dans la figure 9.5, nous nous apercevons qu'elles sont inégalement réparties. Effectivement, `amount` et `loc` sont plutôt sujettes aux délétions, tandis que `org` est soumise à un grand nombre d'erreurs de types. Nous voyons également que `loc` provoque plus d'erreurs d'insertions que les autres types. De manière générale, nous remarquons qu'un travail supplémentaire sur `loc` et `org` (lié, en partie, aux métonymies) est nécessaire pour améliorer les performances globales du système.

### 9.2.4 Comportement du système et expériences supplémentaires

Le modèle à régression logistique donne donc les meilleurs performances globales, par une estimation des probabilités des marqueurs qui tend à équilibrer les erreurs commises et minimiser leur somme pour le SER. Nous cherchons à obtenir plus d'information sur la manière dont les entités nommées sont détectées et reconnues. Nous confrontons alors les marqueurs présents dans l'annotation de référence aux probabilités fournies par le modèle numérique. A une position donnée, nous mettons en place les métriques suivantes :

- **Detect**, taux de détection des marqueurs : accord sur la présence d'un marqueur quelconque d'entité nommée (versus $\emptyset$).
- **Reco**, taux de reconnaissance des marqueurs : pour $N$ marqueurs (potentiellement $\emptyset$) en annotation de référence, combien de marqueurs parmi les $N$ plus probables

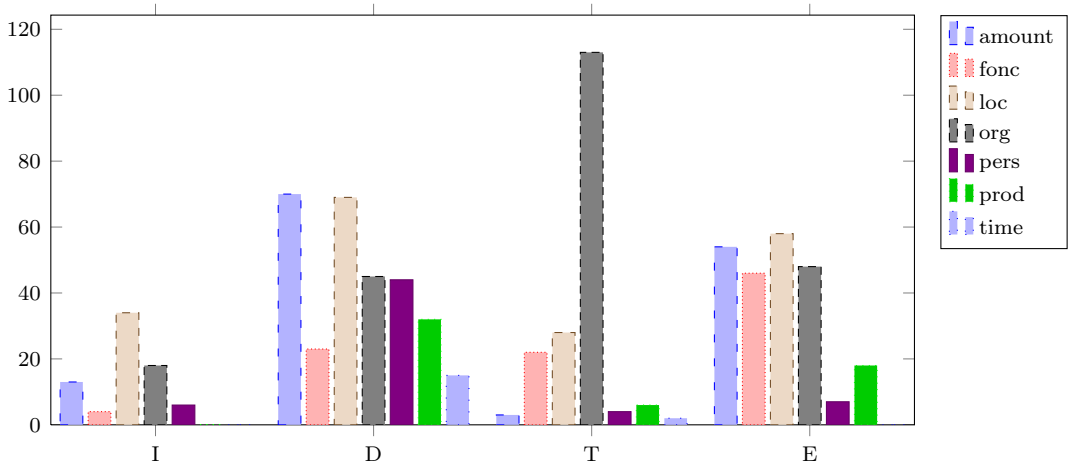



FIGURE 9.5 – Erreurs par type sur Ester2

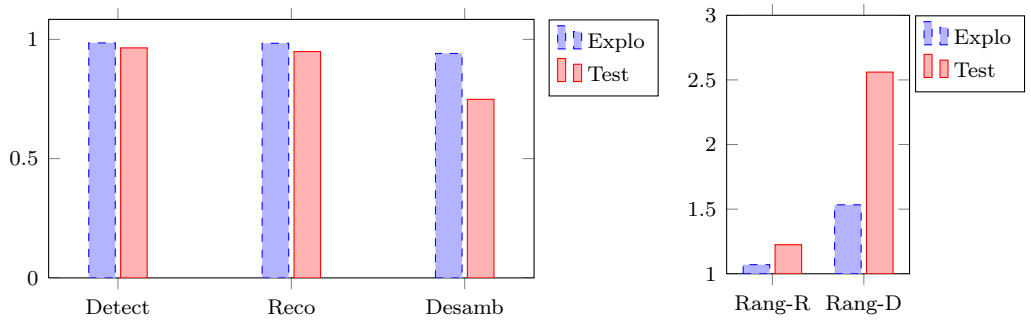
(dont ∅) sont en annotation de référence.

- **Desamb**, taux de désambiguïsation des marqueurs : identique au taux de reconnaissance en excluant les positions sur lesquelles l'annotation de référence et le marqueur le plus probable sont tous deux à ∅.
- **Rang-R**, rang moyen de reconnaissance des séquences : rang de la séquence de l'annotation de référence au sein des séquences produites par le systèmes ordonnées selon leurs probabilités.
- **Rang-D**, rang moyen de désambiguïsation des séquences : identique au rang moyen de reconnaissance des séquences en excluant les positions sur lesquelles l'annotation de référence et le marqueur le plus probable sont tous deux à ∅.

Nous mesurons ces valeurs lorsque les performances globales sont les meilleures en évaluant notre système soit sur le corpus qui a servi à explorer les données, soit sur le corpus de test. La figure 9.6 présente ces résultats. Comme attendu, les taux et les rangs affichent de moins bons résultats lorsque l'on se focalise sur la désambiguïsation des marqueurs ou des séquences de marqueurs. Nous constatons cependant que la dégradation est bien plus forte sur le corpus de tests que sur le corpus exploré : il semble que `mXS` parvient à modéliser efficacement les données qu'il explore, mais qu'il a plus de difficulté à généraliser le modèle sur un autre corpus. Une analyse détaillées de la précision, du rappel et de la confusion sur les marqueurs peut être trouvée en [Nouvel *et al.*, 2011a].

Nous menons également de nombreuses expériences afin de déterminer le comportement de `mXS` dans diverses configurations et de le comparer aux approches orientées données et orientées connaissances que nous sommes en mesure de mettre en œuvre. En voici les descriptions :

- `Logit-Dicos` : identique à `Logit`, mais les ressources lexicales sont désactivées.
- `Logit+Test` : identique à `Logit`, l'exploration des données est conduite sur la fusion des corpus `Ester2-Dev`, `Ester2-Corr` et `Ester2-Held`, donc en connaissance des données sur lesquelles l'approche est évaluée.




(a) Taux de détection, reconnaissance et désambiguïsation (b) Rang moyen des séquences

FIGURE 9.6 – Détection, reconnaissance, désambiguïsation et ordonnancement sur le corpus exploré (Explo) et de test (Test) avec l'approche `Logit` sur Ester2

- `Logit-DXX` : identique à `Logit+Segs`, mais les motifs sont filtrés par utilisation de la relation d'équivalence entre motifs (c.f. 5.5.4) à XX% près.
- `CRF` : approche CRF implémentée à l'aide de l'outil wapiti [2] [Lavergne *et al.*, 2010], pour laquelle les ressources lexicales sont mises à disposition (une observation par trait sémantique).
- `CRF+Test` : identique à `CRF` pour l'approche et à `Logit+Test` pour les données.
- `CasEN` : système CasEN, dans une version améliorée depuis Ester2.
- `Logit+CasEN` : identique à `Logit`, avec l'ajout des entités nommées reconnues par `CasEN` comme niveau supplémentaire de la hiérarchie au dessus des ressources lexicales.

La table 9.2 récapitule les paramètres des approches et le détail résultats qu'elles obtiennent lorsqu'elles maximisent le SER :

Lorsque les ressources lexicales sont désactivées (`Logit-Dicos`), nous constatons une forte recrudescence des erreurs de type. Nous en déduisons que la *reconnaissance* des entités souffre plus de l'absence de ces ressources que leur *détection*. Lorsque nous incluons le corpus de tests dans l'exploration des données (`Logit+Test`), nous voyons que le taux d'erreur est très fortement diminué. Mis à part les erreurs de délétion, le système est alors pénalisé par un nombre proportionnellement assez important d'erreurs d'extension. Comparativement à `CRF+Test`, `Logit+Test` atteint un plus haut niveau de performances. Ceci confirme que le système explore efficacement le corpus d'exploration, mais ajuste probablement ses paramètres de manière trop spécifique (sur-apprentissage).

Les expériences `Logit-DXX`, menées en réduisant le nombre de motifs par ajustement de la granularité des classes d'équivalences montrent que nous sommes en mesure (par rapport à `Logit+Segs`), pour `Logit-D25` et `Logit-D50` de réduire considérablement le nombre de motifs extraits (facteur 2) avec une dégradation remarquablement modérée des performances.

Les expériences `CRF` et `CasEN` nous permettent de situer l'approche par rapport aux systèmes orientés données et connaissances. Le système `CRF` donne de meilleures perfor-

2. `http://wapiti.limsi.fr`

| Approche | F | C | Règles | SER | I | D | T | E | P | R | Fm |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Règles | 03 | 70 | 12 683 | **47,26** | 113 | 753 | 162 | 221 | 84,55 | 52,49 | 0,65 |
| Bayes | 03 | 10 | 58 087 | **38,30** | 215 | 363 | 197 | 289 | 78,55 | 65,11 | 0,71 |
| Logit | 03 | 05 | 59 610 | **27,58** | 75 | 298 | 178 | 231 | 83,07 | 71,00 | 0,77 |
| Logit+Segs | 03 | 05 | 132 205 | **27,97** | 81 | 301 | 184 | 221 | 83,51 | 70,85 | 0,77 |
| Logit-Dicos | 03 | 05 | 51 490 | **33,85** | 97 | 353 | 322 | 181 | 75,56 | 65,90 | 0,70 |
| Logit+Test | 03 | 05 | 70 749 | **11,67** | 35 | 126 | 11 | 133 | 93,56 | 87,55 | 0,90 |
| Logit-D25 | 03 | 05 | 111 489 | **29,35** | 85 | 314 | 181 | 261 | 81,55 | 69,69 | 0,75 |
| Logit-D50 | 03 | 05 | 63 995 | **30,43** | 94 | 316 | 192 | 282 | 80,53 | 68,66 | 0,74 |
| Logit-D75 | 03 | 05 | 55 592 | **36,06** | 96 | 438 | 178 | 282 | 74,44 | 64,50 | 0,69 |
| CRF | na | na | na | **24,64** | 72 | 229 | 183 | 200 | 84,97 | 77,90 | 0,81 |
| CRF+Test | na | na | na | **13,31** | 40 | 101 | 68 | 169 | 92,20 | 89,02 | 0,91 |
| CasEN | na | na | na | **28,58** | 42 | 343 | 165 | 260 | 85,21 | 69,78 | 0,77 |
| Logit+CasEN | 05 | 05 | 38 530 | **24,64** | 79 | 243 | 189 | 191 | 83,21 | 74,40 | 0,79 |

TABLE 9.2 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) des approches aux meilleurs seuils de Fréquence (F) et de Confiance (C)

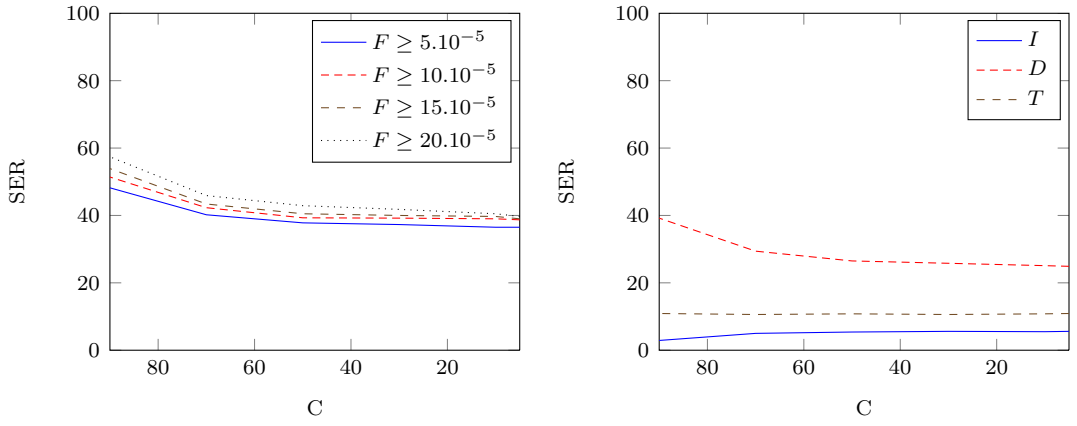
mances, mais il reste difficile de déterminer à quel point cet écart est significatif. Nous remarquons que, parmi les trois systèmes `CasEN` commet le plus d'erreurs de délétion, `CRF` le plus d'erreurs de type, `mXS` le plus d'erreurs d'insertion. Par ailleurs, l'hybridation de `Logit` et `CasEN` en `Logit+CasEN` nous montre qu'il est possible d'atteindre des performances similaires à `CRF` lorsque les données fournies à `mXS` sont enrichies en indices supplémentaires sur les entités nommées [Nouvel *et al.*, 2012].

## 9.3 Etape

### 9.3.1 Régression logistique

Sur le corpus Etape, nous réalisons nos expériences uniquement avec le modèle de régression logistique. Le détail des résultats expérimentaux à divers seuils de fréquence et de confiances sont également fournis en annexe A. La figure 9.7 en présente les résultats, pour lesquels les erreurs sont cette fois exprimées sous forme de taux et les erreurs de type et d'extension sont regroupées comme erreurs de substitution [Galibert *et al.*, 2011]. Nous voyons que le système `Logit` présente un comportement aussi stable que pour Ester2, malgré la plus grande complexité de la tâche (plus de types à considérer, problématique d'imbrication). Nous constatons par ailleurs que l'abaissement de la fréquence améliore significativement les performances. De manière générale, en tenant compte du fait que nous ne descendons pas en dessous d'une fréquence absolue de 13 et que nous n'utilisons pas le corpus `Etape-Quaero` lors de l'exploration, il nous semble que les résultats que nous présentons ici disposent de marges pour l'amélioration des performances.

Comme nous le voyons dans la figure 9.7, les expériences conduites en utilisant les motifs de segments nous donnent des courbes très similaires, mais avec cette fois-ci un SER amélioré, ce gain étant équitablement réparti entre les divers types d'erreurs commises. Par ailleurs, lorsque nous désactivons les ressources lexicales (`Logit-Dicos`), nous constatons




(a) Performances

(b) Erreurs, $F \geq 5.10^{-5}$

FIGURE 9.7 – Performances (SER) erreurs d'Insertion (I), de Délétion (D), de Type (T) selon la Fréquence relative (F) et la Confiance (C) avec l'approche `Logit` sur Etape

comme pour Ester2 une forte augmentation du nombre d'erreurs et du SER (45.2%). La figure 9.8 présente les performances en précision et en rappel avec ces systèmes pour les types les plus fréquents (au moins 100 occurrences au sein de l'évaluation). Nous y voyons alors clairement le gain indiscutable dû à l'utilisation des ressources lexicales, notamment pour `org.ent`, `prod.media`, `loc.adm.nat`, `loc.adm.town`. Les gains par utilisation des motifs de segments portent surtout sur `org.adm` et `loc.adm.town` en précision et `unit` et `org.ent` en rappel.

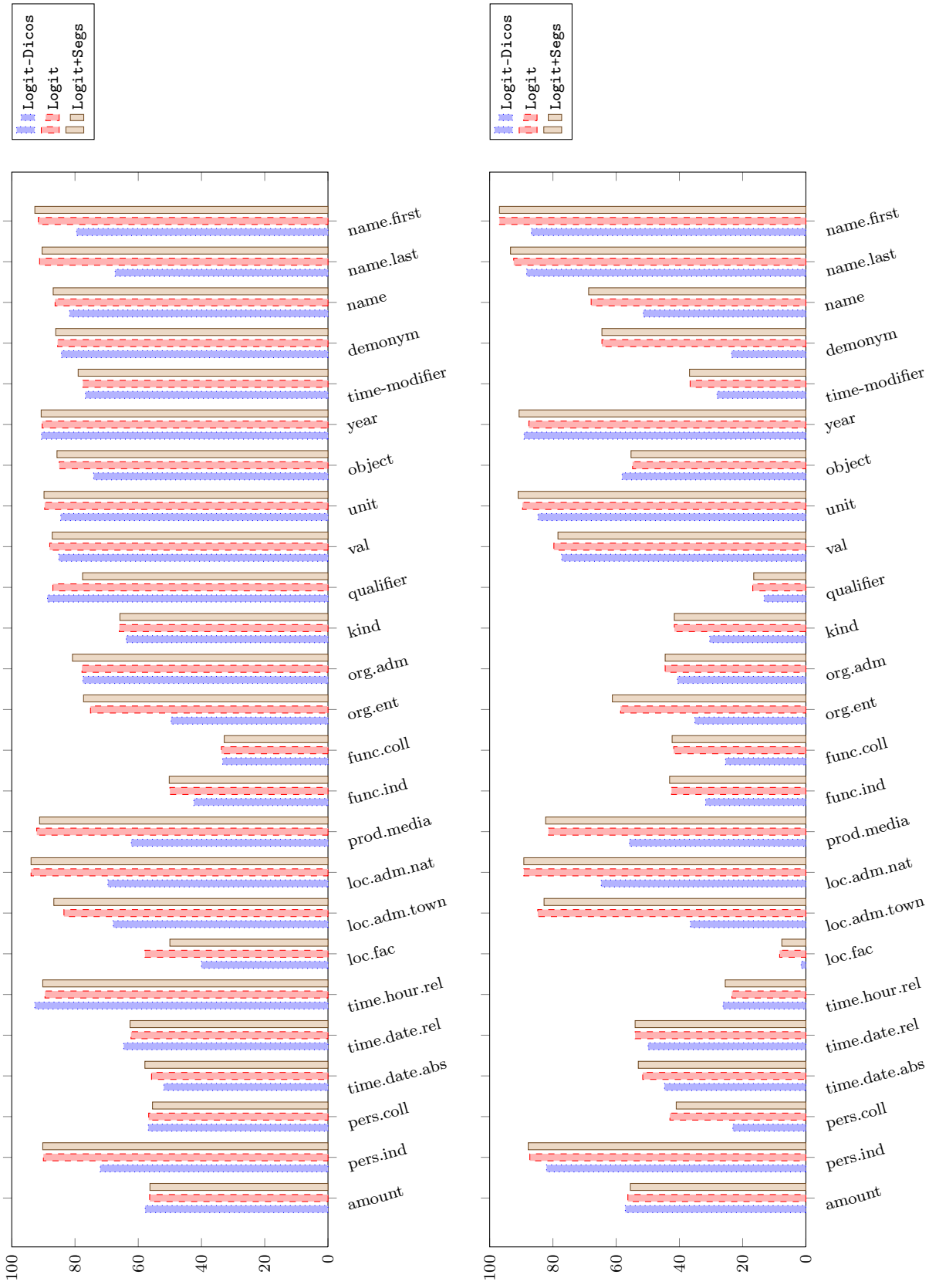
La figure 9.9 donne la précision et le rappel lorsque l'on réalise l'évaluation uniquement sur types principaux d'entités d'Etape. Nous remarquons que, de manière générale, les performances sont moins bonnes que pour la campagne Ester2. Plusieurs types, en particulier les expressions de temps, ont un rappel relativement faible. Nous notons que les montants et les fonctions restent difficile à reconnaître correctement, ce qui pourrait être lié au fait que ces types correspondent souvent à des descriptions définies.

Nous menons des évaluations séparées des types primaires (sans sous-types) d'entités nommées et de composants, que nous comparons avec l'évaluation globale en table 9.3. Il apparaît que les entités nommées sont moins bien reconnues que leurs composants, en particulier du point de vue des erreurs de substitution.

| Types | SER | I | D | S | P | R | Fm |
|---|---|---|---|---|---|---|---|
| **Entités** | **38,9** | 6,9 | 25,4 | 12,3 | 76,4 | 62,3 | 68,6 |
| **Composants** | **33,0** | 4,2 | 25,0 | 6,5 | 86,4 | 68,5 | 76,4 |
| **Test** | **35,9** | 5,6 | 24,2 | 10,8 | 79,8 | 64,9 | 71,6 |

TABLE 9.3 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Substitution (S), Précision (P), Rappel (R), F-mesure (Fm) sur les types primaires et sur toutes les annotations d'Etape

Par ailleurs, lorsque nous analysons plus en détail les évaluation par types d'erreurs




(a) Précision

(b) Rappel



FIGURE 9.8 – Performances par types (Etape)

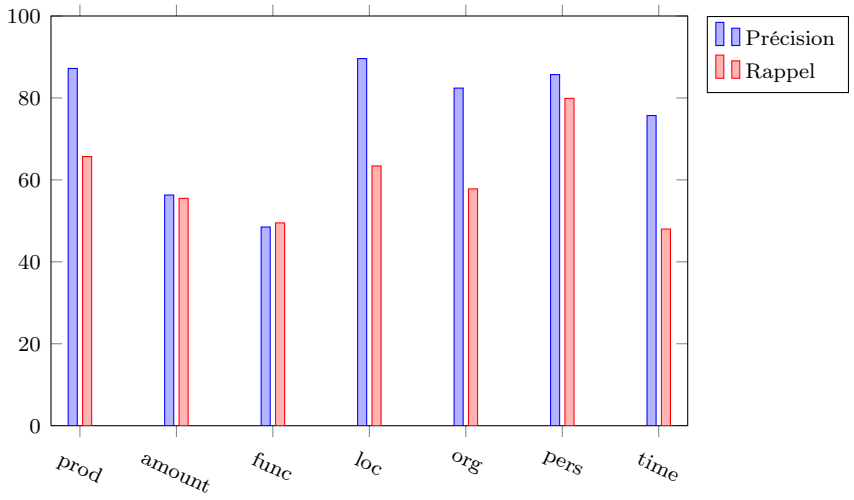
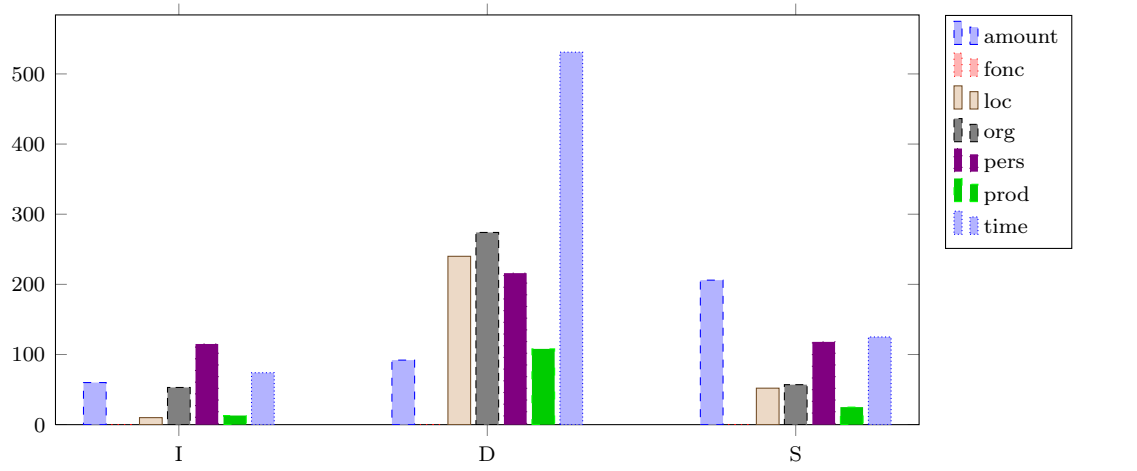



FIGURE 9.9 – Performances par types primaires (Etape)

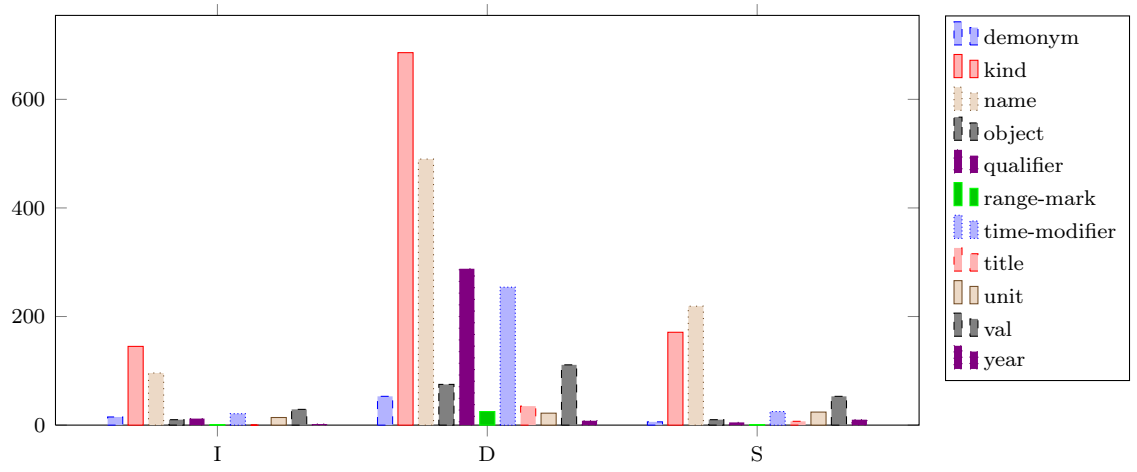
et par types primaires en figure 9.10, nous voyons que les résultats sont assez différents de ceux obtenus sur Ester2. Effectivement, nous obtenons ici un important nombre de délétions pour les `time`, qui apparaît être lié aux difficultés à reconnaître le composant `time-modifier` (*'depuis'*, *'avant'*, *'il y a'*, etc.). Les erreurs de type sur les `org` sont bien moindres que pour Ester2. Nous retrouvons, en substitutions, les erreurs d'extensions pour les `amount` mentionnées pour Ester2. Les entités `org` et `loc` occasionnent, comme pour Ester2, un nombre significatif de délétions, mais dans des proportions différentes.

En ce qui concerne les composants, nous voyons clairement apparaître des difficultés importantes concernant les composant `kind` et `name`, qui paraissent pour partie lié à leur volume (c.f. 7.3.3.2), mais affichent au delà des performances très perfectibles en figure 9.8 (cette figure ne présentant que les types ayant donné lieu à plus de 10 erreurs). Nous remarquons également un nombre important de délétions concernant `qualifier` et `time-modifier`, qui présentent tous deux un rappel assez bas en table 9.8. Cette partie de l'annotation étant plus exploratoire, il paraît difficile de tirer des conclusions dans l'immédiat, et de faire la part des choses entre des réalités linguistiques à préciser et les difficultés de reconnaissance automatique liées aux systèmes.

Dans l'ensemble, il apparaît que la reconnaissance d'entités nommées présente des difficultés supplémentaires sur Etape par rapport à Ester2. Ceci peut être mis sur le compte des données (plus spontanées), mais aussi sur la présence d'entités nommées étendues, construites autour de noms communs, qui recouvrent des expressions linguistiques plus difficiles à reconnaître. Par ailleurs, en incluant les composants, le plus grand nombre d'éléments à reconnaître rend plus complexe l'exploration des données et l'ajustement des paramètres du modèle.




(a) Entités




(b) Composants

FIGURE 9.10 – Erreurs par types primaires d'entités nommées et de composants sur Ester2

### 9.3.2 Comportement du système

Comme pour Ester2, nous menons diverses expériences afin de déterminer quelles sont les performance de `mXS` lorsque l'on désactive les ressources lexicales, ou lorsque l'on ajoute le corpus `Etape-Dev` (sur lequel le système est évalué) lors de l'exploration. Les résultats détaillés des performances obtenues par les diverses configurations sont présentés en table 9.4. Nous y constatons également une forte dégradation des performances pour la configuration `Logit-Dicos` et une importante amélioration pour la configuration `Logit+Test`. Pour les expériences `Logit-DXX`, nous observons, encore plus que pour Ester2, une dégradation des performances assez faible proportionnellement à la réduction du nombre de règles.

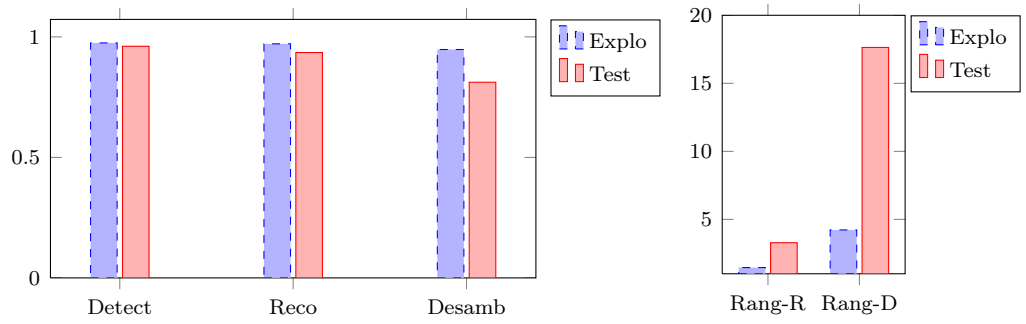
| Approche | F | C | Règles | SER | I | D | S | P | R | Fm |
|---|---|---|---|---|---|---|---|---|---|---|
| `Logit` | 03 | 10 | 106 062 | **36,1** | 5,7 | 24,4 | 10,9 | 79,6 | 64,7 | 71,4 |
| `Logit+Segs` | 03 | 05 | 143 205 | **35,9** | 5,6 | 24,2 | 10,8 | 79,8 | 64,9 | 71,6 |
| `Logit-Dicos` | 03 | 05 | 80 231 | **45,2** | 5,9 | 30,2 | 16,3 | 70,7 | 53,5 | 60,9 |
| `Logit+Test` | 03 | 05 | 141 550 | **26,3** | 3,2 | 18,6 | 8,1 | 86,6 | 73,3 | 79,4 |
| `Logit-D25` | 03 | 05 | 100 027 | **36,2** | 5,6 | 24,6 | 10,9 | 79,7 | 64,6 | 71,3 |
| `Logit-D50` | 03 | 05 | 73 332 | **36,7** | 5,4 | 25,2 | 11,0 | 79,5 | 63,8 | 70,8 |
| `Logit-D75` | 03 | 05 | 50 408 | **39,0** | 5,4 | 27,0 | 11,7 | 78,2 | 61,3 | 68,7 |

TABLE 9.4 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Substitution (S), Précision (P), Rappel (R), F-mesure (Fm) des approches aux meilleurs seuils de Fréquence (F) et de Confiance (C)

Nous mesurons également le comportement de `mXS` selon les mêmes métriques que pour Ester2. Ces résultats sont présentés en figure 9.11. Les taux de détection, de reconnaissance et de désambiguïsation des marqueurs sont similaires à ceux d'Ester2, il ne semble pas y avoir de difficulté supplémentaire de ce point de vue. Nous voyons que le rang moyen des séquences de marqueurs est bien plus élevé : ceci peut être mis sur le compte de la nature du corpus, le nombre de séquences de marqueurs possibles étant bien plus important (475 pour Etape, 88 pour Ester2). Au vu de ces faibles performances locales, nous concluons que calcul de vraisemblance, par combinaison des probabilités sur la séquence, prend une décision globale de qualité en contraignant les décisions locales individuelles. Enfin, nous constatons, comme pour Ester2, une forte dégradation du comportement lorsque l'on passe du corpus d'exploration au corpus de tests.

## 9.4 Discussion

Nous l'avons vu, lorsqu'il s'agit d'utiliser les règles d'annotation explorées pour réaliser la reconnaissance des entités nommées, les meilleurs performances sont obtenues par utilisation d'un modèle probabiliste à base de régression logistique (ou maximum d'entropie). Si ce constat a pu être fait par ailleurs, nous l'appuyons ici sur une extraction de règles avec une exigence d'objectivité et d'exhaustivité : nous ne décidons pas des règles utilisées par `mXS` mais mettons à sa disposition des axes d'analyses, des enrichissements de diverses natures, des métriques pour évaluer l'intérêt des règles. Nous constatons qu'un tel système est alors en mesure d'extraire et de combiner ces indices, même lorsqu'ils sont rares (basse




(a) Taux de détection, reconnaissance et désambiguïsation (b) Rang moyen des séquences

FIGURE 9.11 – Détection, reconnaissance, désambiguïsation et ordonnancement sur le corpus exploré (Explo) et de test (Test) avec l'approche `Logit` (Etape)

fréquence) ou faiblement corrélés aux entités nommées (faible confiance).

Par ailleurs, contrairement aux intuitions que nous pourrions avoir, il apparaît qu'il n'est pas nécessaire de catégoriser *tous* les tokens qui constituent des entités nommées pour obtenir de bonnes performances. En nous focalisant sur les instructions (marqueurs), nous parvenons à reconnaître les entités nommées de manière satisfaisante, et ce sans avoir encore identifié précisément les limites de `mXS` (corpus restreints, fréquence à abaisser). De plus, grâce au lien réalisé entre les marqueurs et l'annotation, l'approche est naturellement conçue pour la reconnaissance des entités nommées structurées (avec imbrications par composants). Il reste à déterminer si d'autres tâches relevant de problématiques similaires peuvent être traitées par une telle approche.

Le constat est plus contrasté en ce qui concerne les motifs de segments. Ils permettent manifestement de réduire la combinatoire lors de l'exploration des données, ce d'autant plus que les données sont enrichies et structurées. Nous avons vu qu'ils étaient peu efficaces pour le corpus Ester2, mais qu'ils permettaient d'améliorer les performances pour le corpus Etape. Nous émettons l'hypothèse que l'abstraction qu'ils introduisent permet de collecter plus de règles d'annotation à divers degrés de granularité, tout en ayant l'inconvénient de décrire moins précisément les niveaux les plus fins. Cette formulation alternative nous paraît cependant intéressante lorsqu'il s'agit d'explorer exhaustivement et en profondeur des données volumineuses tout en limitant l'explosion combinatoire du nombre de motifs à prendre en considération.

# Conclusion et perspectives

Les travaux présentés ici portent sur le traitement automatique des langues, plus précisément à l'intersection entre la linguistique (définition des entités nommées, instructions du langage, marqueurs d'annotation) et l'exploration de données (motifs, motifs de segments, règles d'annotation). Au travers d'une rétrospective, nous donnons des éléments permettant de caractériser les entités nommées. Nous espérons que les propriétés que que nous proposons pour les définir contribuera à affiner leur description et leur annotation. Ensuite, la reconnaissance d'entités nommées est modélisée à l'aide d'instructions locales sous-jacentes. Notre travail repose donc sur la possibilité de décomposer la problématique. En ce sens, nous prenons de la distance par rapport aux approches traditionnelles, qui visent généralement à catégoriser des mots ou des expressions du langage. Au lieu de quoi, nous menons l'exploration des données à un niveau de granularité plus fin, celui des marqueurs (balises) d'annotation, ce qui implique la possibilité de détecter séparément le début et la fin des entités nommées, avant d'en déduire une annotation vraisemblable. Les évaluations nous indiquent que l'approche, quoiqu'encore expérimentale, est compétitive pour reconnaître les entités nommées.

Dans ses fondements, le système élaboré s'inspire à la fois des approches orientées connaissances (transducteurs et ressources lexicales) et orientées données (extraction de règles d'annotation, ajustement de paramètres par régression logistique). Ainsi, nous mettons à disposition un modèle qui permet l'observation du langage naturel (ici les entités nommées) dans sa complexité, au travers du cadre formel proposé pour extraire des motifs. Les études que nous présentons permettent d'émettre et de vérifier des hypothèses pour ce qui concerne la reconnaissance d'entités nommées : égale répartition des motifs selon leur confiance, longueur limitée des motifs, hétérogénéité de certains types d'entités nommées, importance du lexique. Pour la reconnaissance d'entités nommées ou d'autres problématiques similaires, il nous paraît essentiel de disposer de tels outils afin d'étudier le langage naturel dans sa richesse et de manière à la fois exhaustive et objective.

Avec les annotations mises à disposition par les projets Quaero et Etape, les entités nommées sont décomposées en de multiples niveaux d'annotation et nous pouvons alors pleinement considérer la problématique comme une tâche de structuration de données. La finesse des types d'entités nommées et la possibilité de les imbriquer demande à élaborer un modèle d'une plus grande complexité. La représentation des motifs à l'aide de segments donne des résultats satisfaisants dans ce contexte, à la fois pour explorer les données dans des temps raisonnables et pour estimer la vraisemblance des marqueurs. A l'aide d'un modèle linéaire simple, nous sommes en mesure de faire le lien entre les probabilités des marqueurs et celles des séquences de marqueurs. La formalisation devient ainsi adéquate à

la reconnaissance de plusieurs niveaux d'annotation par extraction des règles d'annotation et leur paramétrage au sein d'un modèle à régression logistique à partir de données.

Au travers des expériences, nous relevons la difficulté pour mXS de généraliser depuis le corpus d'où sont extraits les motifs vers le corpus d'évaluation. Ce phénomène, courant pour les approches orientées données, est plus particulièrement saillant lorsque l'on cherche à déterminer le rang moyen des marqueurs corrects lorsqu'ils sont ordonnés selon les probabilités calculées par le système. Il nous semble pertinent d'interroger à ce sujet l'adéquation des algorithmes de classification pour l'approche que nous adoptons. Plus généralement, malgré les restrictions que nous imposons (seuils de fréquence et de confiance, restriction des items lexicaux dans les règles d'annotation, règles d'annotation atomiques), la difficulté reste d'établir un modèle à partir d'un corpus qui soit suffisamment générique pour donner de bonnes performances sur d'autres corpus.

Nous donnons ici quelques pistes pour aller plus loin dans la continuité de ces travaux ou sur des problématiques connexes liées à la reconnaissance d'entités nommées :

**Notion d'entité nommée**

Les éléments que nous apportons en première partie, dans l'objectif de mieux circonscrire la notion d'entité nommée, cherchent à ne pas se fonder sur les applications qui les sollicitent ou les définitions en extension. Nous espérons qu'il sera alors possible de mettre à l'épreuve ces propriétés de stabilité (désignation et référent) et d'opérabilité (vocation à être paramètres de prédications) afin de mieux préciser et définir ce que sont les entités nommées en dehors de contextes particuliers. De ce point de vue, il nous semble utile d'interroger les modes de désignation et les natures des objets référencés selon la sous-tâche en jeu (détection, reconnaissance, résolution) lors de l'étude des entités nommées.

**Paramétrages du système**

La nécessité de développer les algorithmes et structures de données adéquates nous a peu laissé la possibilité d'ajuster finement les paramètres du système et de déterminer les limites de l'approche. Deux axes nous semblent pouvoir nécessiter des approfondissements. En premier lieu, il semble possible, en particulier pour la campagne Etape, d'être en mesure d'exploiter tous les jeux de données à disposition, ce qui nécessite de passer à l'échelle pour plusieurs composants de mXS. Par ailleurs, malgré les nombreuses configurations testées, une étude plus systématique de l'influence des paramètres (notamment pour la régression logistique) pourrait permettre de déterminer plus précisément quel niveau optimal de performance peut être atteint.

**Améliorer le calcul des probabilités locales**

Au delà d'un paramétrage plus fin, il serait certainement avantageux d'intégrer le module qui ajuste ces paramètres au vu des données (régression logistique), peut-être même par des estimations en cours d'exploration. A cet effet, divers autres modèles pourraient se révéler adéquats : réseaux bayésiens, forêts d'arbres décisionnels, réseaux de neurones, fonctions de croyance, etc.

En particulier, il nous parait intéressant de confronter (ou conjuguer) les approches probabilistes (maximiser les probabilités des entités correctes) aux approches par ordonnancement (se limiter à faire en sorte que les entités correctes aient le meilleur rang possible). Quelques expériences préliminaires nous ont montré l'intérêt de tenir compte des

proximités entre marqueurs afin d'optimiser la performance globale : par exemple, à défaut de détecter le début d'une entité de type personne, le début d'une organisation est, a priori, bien moins erroné que la fin d'une expression de temps. En somme, comme cela est souvent le cas dans le cadre de systèmes orientés données, le paramétrage pourrait être affiné en tenant mieux compte de la problématique et des objectifs, dans notre cas la fonction et la nature des marqueurs.

**Ajouter la prise en compte de la vraisemblance globale**

A un plus haut niveau, de nombreuses directions pourraient être explorées, autant dans le cadre de notre approche que plus largement pour d'autres approches qui se focalisent sur les entités nommées. Leur résolution reste une problématique difficile dont l'importance est incontestable pour la recherche d'information. L'interaction entre entités nommées au sein d'ensembles de textes pourrait aider à leur reconnaissance et à leur résolution. De la même manière, la détection et la résolution d'anaphores, la catégorisation verbale, la recherche de cadres de sous-catégorisation sont des pistes qui nous paraissent prometteuses et dont l'interaction avec la reconnaissance d'entités nommées pourraient faire l'objet d'études approfondies.

**Autres problématiques, autres jeux de données**

L'approche que nous avons présentée, si elle est ici dédiée à la reconnaissance d'entités nommées, nous paraît pouvoir être mise en application pour d'autres tâches similaires liées à l'annotation automatique de données. Les expériences préliminaires que nous avons menées en syntaxe (corpus arboré) n'ont pas été suffisamment poussées pour valider ou écarter la pertinence de l'approche pour réaliser une analyse en constituants. Par ailleurs, en dehors du langage naturel, la structuration locale sur segments s'appuyant sur un enrichissement de données pourrait trouver des terrains d'expérimentation assez divers (séquences ADN, logs, etc.). En élargissant, il s'agit de structurer (annoter à plusieurs niveaux) un flux de données (séquence) à l'aide d'indices locaux (reposant sur la structure et l'ontologie). L'originalité de l'approche par rapport à d'autres résidant, outre son cadre formel, dans la détection séparée du début ou de la fin d'éléments (ou segments) à détecter et reconnaître.

# Annexes

# Annexe A

# Tableaux de résultats

**Ester2, approche `Règles`**

Comme le montre le graphique 9.1, l'approche `Règles` demande un paramétrage, en particulier de la confiance.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 947 | 121 | 251 | 345 | 73,80 | 59,15 | 55,56 | 0,57 |
| **03** | **10** | 784 | 138 | 231 | 372 | 66,10 | 63,49 | 57,60 | 0,60 |
| **03** | **30** | 425 | 305 | 188 | 349 | 53,13 | 72,37 | 58,53 | 0,65 |
| **03** | **50** | 244 | 517 | 189 | 294 | 49,78 | 77,03 | 53,97 | 0,63 |
| **03** | **70** | 113 | 753 | 162 | 221 | 47,26 | 84,55 | 52,49 | 0,65 |
| **03** | **90** | 43 | 1143 | 95 | 149 | 54,72 | 90,51 | 47,00 | 0,62 |
| **05** | **05** | 898 | 148 | 250 | 341 | 72,60 | 59,64 | 55,03 | 0,57 |
| **05** | **10** | 758 | 178 | 229 | 370 | 66,18 | 63,89 | 56,71 | 0,60 |
| **05** | **30** | 414 | 359 | 180 | 334 | 54,25 | 72,73 | 56,87 | 0,64 |
| **05** | **50** | 239 | 569 | 185 | 287 | 51,15 | 77,27 | 52,49 | 0,63 |
| **05** | **70** | 102 | 830 | 152 | 217 | 49,36 | 85,20 | 50,32 | 0,63 |
| **05** | **90** | 30 | 1251 | 81 | 138 | 57,78 | 91,37 | 44,37 | 0,60 |
| **07** | **05** | 854 | 190 | 245 | 330 | 72,20 | 59,72 | 53,31 | 0,56 |
| **07** | **10** | 704 | 230 | 224 | 356 | 65,61 | 64,35 | 54,47 | 0,59 |
| **07** | **30** | 406 | 438 | 173 | 326 | 56,66 | 72,72 | 54,49 | 0,62 |
| **07** | **50** | 234 | 652 | 178 | 281 | 53,75 | 77,31 | 50,42 | 0,61 |
| **07** | **70** | 101 | 928 | 144 | 201 | 52,46 | 85,40 | 47,56 | 0,61 |
| **07** | **90** | 28 | 1351 | 78 | 127 | 61,04 | 91,83 | 41,05 | 0,57 |
| **10** | **05** | 808 | 201 | 245 | 324 | 70,69 | 60,07 | 52,99 | 0,56 |
| **10** | **10** | 691 | 266 | 222 | 349 | 65,82 | 64,79 | 54,08 | 0,59 |
| **10** | **30** | 399 | 507 | 171 | 306 | 58,64 | 71,79 | 51,00 | 0,60 |
| **10** | **50** | 225 | 745 | 167 | 266 | 56,41 | 76,86 | 46,58 | 0,58 |
| **10** | **70** | 97 | 1021 | 138 | 193 | 55,64 | 84,57 | 43,19 | 0,57 |
| **10** | **90** | 21 | 1471 | 72 | 120 | 65,09 | 92,10 | 36,80 | 0,53 |
| **15** | **05** | 764 | 255 | 239 | 317 | 70,16 | 61,00 | 51,61 | 0,56 |
| **15** | **10** | 687 | 325 | 218 | 329 | 67,37 | 64,01 | 51,62 | 0,57 |
| **15** | **30** | 384 | 591 | 168 | 275 | 60,48 | 71,05 | 47,71 | 0,57 |
| **15** | **50** | 217 | 802 | 161 | 264 | 57,97 | 76,78 | 44,60 | 0,56 |
| **15** | **70** | 94 | 1072 | 126 | 186 | 57,12 | 84,86 | 42,09 | 0,56 |
| **15** | **90** | 18 | 1558 | 64 | 109 | 67,86 | 92,72 | 35,33 | 0,51 |
| **20** | **05** | 764 | 275 | 239 | 315 | 70,89 | 60,56 | 50,66 | 0,55 |
| **20** | **10** | 682 | 352 | 212 | 326 | 68,17 | 63,86 | 50,54 | 0,56 |
| **20** | **30** | 374 | 636 | 166 | 260 | 61,76 | 70,77 | 45,66 | 0,56 |
| **20** | **50** | 195 | 894 | 155 | 232 | 59,70 | 77,12 | 41,91 | 0,54 |
| **20** | **70** | 91 | 1121 | 122 | 175 | 58,51 | 84,89 | 40,65 | 0,55 |
| **20** | **90** | 11 | 1633 | 62 | 102 | 70,27 | 93,08 | 33,46 | 0,49 |
| **inf** | **inf** | 0 | 2501 | 0 | 0 | 99,52 | 100,00 | 18,42 | 0,31 |

TABLE A.1 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Règles` selon la Fréquence (F) et la Confiance (C)

**Ester2, approche** `Bayes`

La figure 9.2 l'indique : l'approche bayésienne permet de mieux combiner les règles extraites, mais un paramétrage de la confiance reste nécessaire.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 134 | 555 | 174 | 255 | 40,88 | 80,47 | 59,37 | 0,68 |
| **03** | **10** | 215 | 363 | 197 | 289 | 38,30 | 78,55 | 65,11 | 0,71 |
| **03** | **30** | 385 | 215 | 211 | 321 | 41,40 | 74,22 | 68,01 | 0,71 |
| **03** | **50** | 314 | 336 | 187 | 292 | 42,00 | 75,71 | 65,41 | 0,70 |
| **03** | **70** | 135 | 581 | 167 | 234 | 41,93 | 80,07 | 59,84 | 0,68 |
| **03** | **90** | 46 | 965 | 95 | 186 | 49,44 | 85,47 | 52,48 | 0,65 |
| **05** | **05** | 126 | 579 | 171 | 258 | 41,44 | 81,23 | 58,52 | 0,68 |
| **05** | **10** | 196 | 395 | 199 | 279 | 38,91 | 78,24 | 63,39 | 0,70 |
| **05** | **30** | 376 | 242 | 207 | 326 | 42,02 | 74,79 | 66,56 | 0,70 |
| **05** | **50** | 295 | 382 | 187 | 291 | 42,94 | 75,40 | 63,42 | 0,69 |
| **05** | **70** | 120 | 635 | 158 | 233 | 42,87 | 80,93 | 58,66 | 0,68 |
| **05** | **90** | 36 | 1068 | 84 | 171 | 52,12 | 87,13 | 49,95 | 0,63 |
| **07** | **05** | 122 | 622 | 174 | 255 | 43,49 | 80,26 | 55,51 | 0,66 |
| **07** | **10** | 196 | 431 | 199 | 283 | 40,53 | 76,92 | 61,86 | 0,69 |
| **07** | **30** | 356 | 279 | 212 | 348 | 43,43 | 73,99 | 64,44 | 0,69 |
| **07** | **50** | 278 | 421 | 180 | 307 | 43,97 | 75,05 | 61,95 | 0,68 |
| **07** | **70** | 112 | 725 | 149 | 220 | 45,61 | 79,56 | 55,83 | 0,66 |
| **07** | **90** | 35 | 1174 | 80 | 160 | 55,84 | 86,21 | 46,56 | 0,60 |
| **10** | **05** | 125 | 671 | 180 | 256 | 45,92 | 79,37 | 52,42 | 0,63 |
| **10** | **10** | 190 | 482 | 202 | 294 | 42,96 | 75,67 | 58,26 | 0,66 |
| **10** | **30** | 346 | 318 | 227 | 349 | 45,07 | 73,42 | 62,14 | 0,67 |
| **10** | **50** | 271 | 498 | 169 | 314 | 46,70 | 73,90 | 58,90 | 0,66 |
| **10** | **70** | 99 | 815 | 146 | 213 | 48,25 | 80,04 | 51,91 | 0,63 |
| **10** | **90** | 26 | 1296 | 74 | 171 | 59,99 | 86,55 | 41,75 | 0,56 |
| **15** | **05** | 122 | 715 | 171 | 273 | 47,66 | 79,39 | 50,30 | 0,62 |
| **15** | **10** | 194 | 556 | 189 | 301 | 46,08 | 74,81 | 54,63 | 0,63 |
| **15** | **30** | 320 | 387 | 218 | 347 | 46,62 | 72,44 | 58,68 | 0,65 |
| **15** | **50** | 252 | 552 | 169 | 313 | 48,21 | 74,00 | 55,98 | 0,64 |
| **15** | **70** | 89 | 901 | 134 | 199 | 50,67 | 80,04 | 48,63 | 0,61 |
| **15** | **90** | 22 | 1382 | 64 | 153 | 62,58 | 87,59 | 39,39 | 0,54 |
| **20** | **05** | 116 | 778 | 174 | 254 | 49,55 | 78,93 | 47,30 | 0,59 |
| **20** | **10** | 189 | 617 | 189 | 279 | 48,01 | 74,25 | 51,80 | 0,61 |
| **20** | **30** | 300 | 436 | 217 | 321 | 47,57 | 72,24 | 56,13 | 0,63 |
| **20** | **50** | 233 | 658 | 167 | 291 | 51,14 | 72,41 | 51,19 | 0,60 |
| **20** | **70** | 89 | 980 | 133 | 185 | 53,43 | 79,56 | 45,25 | 0,58 |
| **20** | **90** | 11 | 1459 | 63 | 156 | 65,00 | 88,12 | 37,54 | 0,53 |
| **inf** | **inf** | 0 | 2501 | 0 | 0 | 99,52 | 100,00 | 18,42 | 0,31 |

TABLE A.2 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Bayes` selon la Fréquence (F) et la Confiance (C)

**Ester2, approche `Logit`**

La figure 9.3 le montre : la régression logistique combine mieux, sans nécessité de paramétrage, les règles extraites. Pour Ester2, c'est dans cette configuration que mXS obtient les meilleures performances.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 75 | 298 | 178 | 231 | 27,58 | 83,07 | 71,00 | 0,77 |
| **03** | **10** | 76 | 313 | 180 | 224 | 28,20 | 83,88 | 69,99 | 0,76 |
| **03** | **30** | 80 | 326 | 185 | 241 | 29,40 | 82,29 | 69,15 | 0,75 |
| **03** | **50** | 78 | 384 | 174 | 265 | 31,99 | 77,29 | 67,63 | 0,72 |
| **03** | **70** | 77 | 479 | 161 | 273 | 35,50 | 74,57 | 65,66 | 0,70 |
| **03** | **90** | 81 | 590 | 130 | 301 | 41,07 | 67,50 | 62,58 | 0,65 |
| **05** | **05** | 96 | 301 | 193 | 232 | 28,74 | 81,59 | 69,74 | 0,75 |
| **05** | **10** | 92 | 301 | 185 | 241 | 28,79 | 81,43 | 69,70 | 0,75 |
| **05** | **30** | 81 | 335 | 185 | 236 | 29,78 | 79,73 | 68,45 | 0,74 |
| **05** | **50** | 84 | 395 | 188 | 266 | 32,91 | 75,97 | 66,33 | 0,71 |
| **05** | **70** | 73 | 473 | 162 | 288 | 35,75 | 74,22 | 64,96 | 0,69 |
| **05** | **90** | 86 | 582 | 134 | 319 | 41,95 | 63,67 | 62,05 | 0,63 |
| **07** | **05** | 103 | 310 | 194 | 235 | 29,54 | 79,38 | 69,93 | 0,74 |
| **07** | **10** | 87 | 308 | 194 | 250 | 29,23 | 81,22 | 69,35 | 0,75 |
| **07** | **30** | 90 | 343 | 184 | 248 | 31,03 | 77,37 | 67,74 | 0,72 |
| **07** | **50** | 90 | 398 | 183 | 267 | 33,64 | 73,81 | 66,44 | 0,70 |
| **07** | **70** | 86 | 485 | 162 | 315 | 37,39 | 72,39 | 64,71 | 0,68 |
| **07** | **90** | 96 | 611 | 129 | 360 | 44,21 | 61,22 | 61,26 | 0,61 |
| **10** | **05** | 90 | 306 | 192 | 260 | 29,71 | 78,16 | 68,89 | 0,73 |
| **10** | **10** | 89 | 311 | 189 | 269 | 29,88 | 79,57 | 68,75 | 0,74 |
| **10** | **30** | 86 | 338 | 184 | 271 | 31,30 | 75,85 | 67,18 | 0,71 |
| **10** | **50** | 103 | 397 | 168 | 294 | 34,38 | 72,50 | 65,92 | 0,69 |
| **10** | **70** | 82 | 508 | 156 | 323 | 38,72 | 70,59 | 62,66 | 0,66 |
| **10** | **90** | 122 | 622 | 132 | 392 | 46,77 | 55,08 | 58,61 | 0,57 |
| **15** | **05** | 107 | 303 | 204 | 268 | 31,14 | 76,25 | 67,57 | 0,72 |
| **15** | **10** | 103 | 300 | 204 | 274 | 31,15 | 76,80 | 67,34 | 0,72 |
| **15** | **30** | 104 | 351 | 196 | 289 | 33,33 | 74,08 | 65,43 | 0,69 |
| **15** | **50** | 100 | 398 | 181 | 326 | 35,36 | 72,51 | 64,64 | 0,68 |
| **15** | **70** | 86 | 540 | 146 | 334 | 40,31 | 67,73 | 61,15 | 0,64 |
| **15** | **90** | 125 | 659 | 130 | 394 | 48,48 | 53,41 | 56,88 | 0,55 |
| **20** | **05** | 105 | 310 | 206 | 279 | 31,60 | 74,57 | 67,25 | 0,71 |
| **20** | **10** | 104 | 302 | 206 | 283 | 31,68 | 75,98 | 67,13 | 0,71 |
| **20** | **30** | 100 | 339 | 207 | 301 | 33,43 | 72,39 | 64,50 | 0,68 |
| **20** | **50** | 117 | 419 | 181 | 347 | 37,66 | 68,49 | 63,18 | 0,66 |
| **20** | **70** | 89 | 578 | 144 | 345 | 42,09 | 67,72 | 59,31 | 0,63 |
| **20** | **90** | 129 | 668 | 135 | 394 | 49,40 | 54,13 | 56,41 | 0,55 |
| **inf** | **inf** | 275 | 580 | 78 | 655 | 60,86 | 31,63 | 53,10 | 0,40 |

TABLE A.3 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit` selon la Fréquence (F) et la Confiance (C)

**Ester2, approche** `Logit+Segs`

Les résultats correspondant à l'approche par segments présentent une performance très comparable à l'approche par des motifs plus classiques. Nous remarquons que s'il y a plus d'erreurs d'insertions, de délétion et de type, il y en a cependant moins en extension : il n'est pas gênant de s'appuyer sur des segments pour délimiter les entités nommées. Par ailleurs, lors que l'on augmente les seuils de fréquence et de confiance, cette approche donne de meilleurs résulats.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 81 | 301 | 184 | 221 | 27,97 | 83,51 | 70,85 | 0,77 |
| **03** | **10** | 80 | 301 | 184 | 224 | 28,07 | 83,88 | 70,71 | 0,77 |
| **03** | **30** | 86 | 322 | 182 | 250 | 29,37 | 81,79 | 69,51 | 0,75 |
| **03** | **50** | 72 | 376 | 178 | 274 | 31,39 | 78,65 | 67,63 | 0,73 |
| **03** | **70** | 72 | 487 | 161 | 266 | 35,25 | 77,00 | 66,03 | 0,71 |
| **03** | **90** | 82 | 609 | 136 | 287 | 40,87 | 69,24 | 63,11 | 0,66 |
| **05** | **05** | 99 | 302 | 183 | 240 | 29,22 | 81,59 | 69,98 | 0,75 |
| **05** | **10** | 84 | 303 | 188 | 252 | 28,92 | 82,10 | 69,40 | 0,75 |
| **05** | **30** | 86 | 323 | 194 | 247 | 29,65 | 79,88 | 68,87 | 0,74 |
| **05** | **50** | 82 | 377 | 180 | 264 | 31,95 | 76,47 | 66,92 | 0,71 |
| **05** | **70** | 73 | 488 | 161 | 285 | 35,68 | 75,39 | 65,75 | 0,70 |
| **05** | **90** | 80 | 590 | 126 | 319 | 41,11 | 67,26 | 63,23 | 0,65 |
| **07** | **05** | 90 | 303 | 198 | 241 | 29,19 | 80,19 | 69,72 | 0,75 |
| **07** | **10** | 88 | 303 | 206 | 252 | 29,69 | 79,69 | 69,01 | 0,74 |
| **07** | **30** | 85 | 327 | 196 | 270 | 30,65 | 78,09 | 67,82 | 0,73 |
| **07** | **50** | 86 | 385 | 184 | 269 | 32,83 | 74,45 | 66,76 | 0,70 |
| **07** | **70** | 82 | 497 | 162 | 304 | 36,99 | 73,69 | 65,65 | 0,69 |
| **07** | **90** | 89 | 619 | 127 | 342 | 42,96 | 64,38 | 62,05 | 0,63 |
| **10** | **05** | 93 | 300 | 202 | 255 | 29,69 | 78,41 | 68,68 | 0,73 |
| **10** | **10** | 95 | 300 | 198 | 271 | 30,37 | 76,93 | 68,28 | 0,72 |
| **10** | **30** | 89 | 334 | 199 | 268 | 31,36 | 76,33 | 66,84 | 0,71 |
| **10** | **50** | 98 | 391 | 177 | 283 | 33,85 | 73,58 | 65,83 | 0,69 |
| **10** | **70** | 94 | 529 | 155 | 333 | 39,65 | 69,64 | 62,86 | 0,66 |
| **10** | **90** | 103 | 642 | 130 | 343 | 45,07 | 60,46 | 58,84 | 0,60 |
| **15** | **05** | 101 | 293 | 217 | 263 | 30,49 | 76,13 | 67,56 | 0,72 |
| **15** | **10** | 110 | 290 | 216 | 277 | 31,16 | 75,52 | 67,36 | 0,71 |
| **15** | **30** | 105 | 332 | 199 | 288 | 32,59 | 74,24 | 65,89 | 0,70 |
| **15** | **50** | 100 | 407 | 175 | 313 | 34,81 | 73,90 | 65,14 | 0,69 |
| **15** | **70** | 98 | 555 | 156 | 344 | 41,28 | 67,28 | 60,83 | 0,64 |
| **15** | **90** | 105 | 669 | 128 | 345 | 46,48 | 59,20 | 57,65 | 0,58 |
| **20** | **05** | 120 | 306 | 207 | 267 | 32,01 | 76,03 | 67,35 | 0,71 |
| **20** | **10** | 113 | 297 | 208 | 285 | 32,28 | 74,29 | 66,34 | 0,70 |
| **20** | **30** | 103 | 328 | 200 | 304 | 33,28 | 72,00 | 64,83 | 0,68 |
| **20** | **50** | 114 | 405 | 184 | 348 | 36,67 | 69,56 | 63,84 | 0,67 |
| **20** | **70** | 98 | 572 | 153 | 343 | 42,10 | 67,38 | 59,80 | 0,63 |
| **20** | **90** | 109 | 674 | 128 | 356 | 47,19 | 58,74 | 57,24 | 0,58 |
| **inf** | **inf** | 275 | 580 | 78 | 655 | 60,86 | 31,63 | 53,10 | 0,40 |

TABLE A.4 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit+Segs` selon la Fréquence (F) et la Confiance (C)

**Ester2, approche** `Logit-Dicos`

Comme attendu, la désactivation des ressources lexicales dégrade considérablement les résultats.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 97 | 353 | 322 | 181 | 33,85 | 75,56 | 65,90 | 0,70 |
| **03** | **10** | 108 | 346 | 324 | 210 | 34,96 | 73,42 | 64,86 | 0,69 |
| **03** | **30** | 104 | 337 | 316 | 222 | 34,88 | 73,84 | 64,73 | 0,69 |
| **03** | **50** | 116 | 446 | 309 | 232 | 39,59 | 69,79 | 63,26 | 0,66 |
| **03** | **70** | 116 | 648 | 259 | 279 | 46,68 | 63,85 | 59,86 | 0,62 |
| **05** | **05** | 116 | 360 | 334 | 204 | 35,66 | 72,22 | 64,80 | 0,68 |
| **05** | **10** | 123 | 349 | 340 | 222 | 35,95 | 72,37 | 64,29 | 0,68 |
| **05** | **30** | 121 | 332 | 332 | 245 | 35,91 | 72,77 | 64,45 | 0,68 |
| **05** | **50** | 114 | 457 | 315 | 259 | 40,66 | 68,65 | 62,03 | 0,65 |
| **05** | **70** | 113 | 641 | 258 | 295 | 46,91 | 62,20 | 59,58 | 0,61 |
| **07** | **05** | 137 | 347 | 344 | 217 | 36,66 | 69,57 | 64,60 | 0,67 |
| **07** | **10** | 137 | 343 | 339 | 232 | 36,67 | 70,21 | 64,55 | 0,67 |
| **07** | **30** | 131 | 344 | 336 | 262 | 37,46 | 68,24 | 63,10 | 0,66 |
| **07** | **50** | 120 | 460 | 317 | 273 | 41,70 | 66,26 | 61,01 | 0,64 |
| **07** | **70** | 122 | 688 | 241 | 313 | 49,58 | 58,22 | 57,91 | 0,58 |
| **10** | **05** | 132 | 322 | 325 | 245 | 36,29 | 69,31 | 65,06 | 0,67 |
| **10** | **10** | 123 | 346 | 334 | 260 | 37,22 | 69,39 | 63,26 | 0,66 |
| **10** | **30** | 128 | 353 | 316 | 266 | 37,60 | 69,41 | 62,18 | 0,66 |
| **10** | **50** | 126 | 471 | 317 | 302 | 43,01 | 64,12 | 59,57 | 0,62 |
| **10** | **70** | 137 | 695 | 246 | 336 | 51,60 | 54,65 | 56,60 | 0,56 |
| **15** | **05** | 146 | 339 | 349 | 279 | 38,48 | 67,98 | 63,37 | 0,66 |
| **15** | **10** | 146 | 346 | 342 | 297 | 39,01 | 66,36 | 62,06 | 0,64 |
| **15** | **30** | 149 | 377 | 332 | 299 | 40,51 | 66,54 | 59,86 | 0,63 |
| **15** | **50** | 132 | 532 | 336 | 313 | 45,71 | 63,12 | 57,01 | 0,60 |
| **15** | **70** | 155 | 768 | 225 | 328 | 54,95 | 50,75 | 53,86 | 0,52 |
| **20** | **05** | 154 | 345 | 331 | 290 | 38,87 | 66,35 | 63,03 | 0,65 |
| **20** | **10** | 146 | 359 | 346 | 311 | 39,77 | 66,27 | 61,79 | 0,64 |
| **20** | **30** | 149 | 394 | 333 | 315 | 41,30 | 65,70 | 59,37 | 0,62 |
| **20** | **50** | 140 | 559 | 315 | 322 | 46,98 | 62,71 | 56,24 | 0,59 |
| **20** | **70** | 163 | 805 | 199 | 344 | 56,62 | 49,23 | 53,20 | 0,51 |
| **inf** | **inf** | 299 | 807 | 82 | 618 | 69,61 | 30,76 | 50,07 | 0,38 |

TABLE A.5 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit-Dicos` selon la Fréquence (F) et la Confiance (C)

**Ester2, approche** `Logit+Test`

Lorsque le système exploite les données sur lesquelles il sera évalué, ses performances s'accroissent très significativement.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 35 | 126 | 11 | 133 | 11,67 | 93,56 | 87,55 | 0,90 |
| **03** | **10** | 40 | 127 | 12 | 142 | 12,19 | 93,50 | 87,39 | 0,90 |
| **03** | **30** | 38 | 158 | 15 | 157 | 13,95 | 93,05 | 84,62 | 0,89 |
| **03** | **50** | 39 | 198 | 22 | 169 | 15,77 | 91,52 | 83,08 | 0,87 |
| **03** | **70** | 34 | 310 | 29 | 202 | 21,07 | 87,98 | 78,68 | 0,83 |
| **03** | **90** | 45 | 421 | 37 | 250 | 28,32 | 78,10 | 74,51 | 0,76 |
| **05** | **05** | 43 | 145 | 18 | 163 | 13,65 | 92,10 | 85,55 | 0,89 |
| **05** | **10** | 43 | 158 | 23 | 176 | 14,60 | 91,75 | 84,39 | 0,88 |
| **05** | **30** | 46 | 191 | 29 | 184 | 16,73 | 90,03 | 82,04 | 0,86 |
| **05** | **50** | 48 | 223 | 32 | 197 | 18,29 | 89,22 | 80,81 | 0,85 |
| **05** | **70** | 36 | 340 | 39 | 245 | 23,48 | 86,56 | 77,42 | 0,82 |
| **05** | **90** | 51 | 452 | 52 | 266 | 30,72 | 75,29 | 72,43 | 0,74 |
| **07** | **05** | 51 | 162 | 31 | 160 | 15,20 | 90,84 | 84,02 | 0,87 |
| **07** | **10** | 56 | 176 | 36 | 177 | 16,45 | 90,11 | 82,70 | 0,86 |
| **07** | **30** | 47 | 212 | 36 | 184 | 17,97 | 88,71 | 80,15 | 0,84 |
| **07** | **50** | 56 | 258 | 36 | 206 | 20,51 | 87,20 | 78,59 | 0,83 |
| **07** | **70** | 43 | 384 | 48 | 258 | 26,45 | 83,72 | 73,93 | 0,79 |
| **07** | **90** | 71 | 498 | 60 | 310 | 35,27 | 68,34 | 68,99 | 0,69 |
| **10** | **05** | 61 | 183 | 48 | 193 | 17,46 | 88,60 | 81,74 | 0,85 |
| **10** | **10** | 55 | 190 | 51 | 215 | 18,19 | 88,10 | 80,22 | 0,84 |
| **10** | **30** | 60 | 221 | 47 | 218 | 19,71 | 87,97 | 78,61 | 0,83 |
| **10** | **50** | 65 | 278 | 47 | 233 | 22,65 | 85,26 | 76,62 | 0,81 |
| **10** | **70** | 48 | 428 | 56 | 271 | 29,10 | 80,86 | 71,27 | 0,76 |
| **10** | **90** | 87 | 520 | 72 | 353 | 38,10 | 63,65 | 66,66 | 0,65 |
| **15** | **05** | 77 | 202 | 61 | 217 | 19,98 | 85,82 | 79,32 | 0,82 |
| **15** | **10** | 71 | 206 | 67 | 237 | 20,66 | 84,91 | 77,23 | 0,81 |
| **15** | **30** | 74 | 235 | 62 | 243 | 21,96 | 85,02 | 76,39 | 0,80 |
| **15** | **50** | 68 | 298 | 57 | 268 | 24,82 | 82,00 | 74,37 | 0,78 |
| **15** | **70** | 67 | 450 | 71 | 310 | 32,14 | 75,15 | 68,49 | 0,72 |
| **15** | **90** | 114 | 559 | 67 | 371 | 41,21 | 58,67 | 63,99 | 0,61 |
| **20** | **05** | 89 | 213 | 70 | 241 | 21,85 | 83,71 | 77,51 | 0,80 |
| **20** | **10** | 88 | 223 | 73 | 252 | 22,80 | 82,01 | 75,65 | 0,79 |
| **20** | **30** | 88 | 247 | 72 | 274 | 24,06 | 82,00 | 74,56 | 0,78 |
| **20** | **50** | 81 | 335 | 69 | 303 | 27,86 | 78,08 | 71,24 | 0,75 |
| **20** | **70** | 74 | 494 | 68 | 314 | 34,33 | 74,74 | 66,17 | 0,70 |
| **20** | **90** | 125 | 578 | 69 | 375 | 43,08 | 57,06 | 62,90 | 0,60 |
| **inf** | **inf** | 263 | 493 | 61 | 669 | 54,94 | 33,51 | 57,70 | 0,42 |

TABLE A.6 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit+Test` selon la Fréquence (F) et la Confiance (C)

**Ester2, approches `Logit-D25`, `Logit-D50` et `Logit-D75`**

Il reste possible, lorsque l'on filtre le nombre de règles extraites, de conserver d'assez bonnes performances. Nous remarquons en particulier une faible augmentation des erreurs de type.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 85 | 314 | 181 | 261 | 29,35 | 81,55 | 69,69 | 0,75 |

TABLE A.7 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit-D25` selon la Fréquence (F) et la Confiance (C)

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 94 | 316 | 192 | 282 | 30,43 | 80,53 | 68,66 | 0,74 |

TABLE A.8 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit-D50` selon la Fréquence (F) et la Confiance (C)

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 96 | 438 | 178 | 282 | 36,06 | 74,44 | 64,50 | 0,69 |

TABLE A.9 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit-D75` selon la Fréquence (F) et la Confiance (C)

**Ester2, approche `CRF`**

L'approche état de l'art reste très performante. Nous voyons ici que ce qui pénalise le plus le système est son silence (rappel) : la préférence est donnée à ne pas annoter, plutôt que de provoquer des erreurs d'insertion ou de type.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| na | na | 72 | 229 | 183 | 200 | 24,64 | 84,97 | 77,90 | 0,81 |

TABLE A.10 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `CRF` selon la Fréquence (F) et la Confiance (C)

**Ester2, approche** `Logit+CasEN`

L'hybridation avec le système `CasEN` nous permet d'obtenir une performance équivalente à l'état de l'art en SER, même si la f-mesure reste inférieure. Comparativement, nous notons un nombre plus important d'erreurs de délétions, mais moins d'erreurs de type.

| F | C | I | D | T | E | SER | P | R | F |
|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 75 | 255 | 192 | 185 | 24,76 | 84,64 | 74,00 | 0,79 |
| **03** | **10** | 77 | 262 | 187 | 194 | 25,05 | 84,51 | 73,87 | 0,79 |
| **03** | **30** | 67 | 268 | 180 | 208 | 25,12 | 85,51 | 73,23 | 0,79 |
| **03** | **50** | 74 | 293 | 191 | 204 | 26,39 | 83,96 | 73,52 | 0,78 |
| **03** | **70** | 82 | 310 | 183 | 203 | 27,46 | 82,42 | 73,49 | 0,78 |
| **03** | **90** | 89 | 335 | 182 | 208 | 28,67 | 83,28 | 73,36 | 0,78 |
| **05** | **05** | 79 | 243 | 189 | 191 | 24,64 | 83,21 | 74,40 | 0,79 |
| **05** | **10** | 79 | 252 | 190 | 208 | 25,13 | 83,49 | 74,14 | 0,79 |
| **05** | **30** | 76 | 255 | 197 | 213 | 25,23 | 83,67 | 73,74 | 0,78 |
| **05** | **50** | 75 | 284 | 194 | 219 | 26,57 | 83,61 | 73,23 | 0,78 |
| **05** | **70** | 88 | 301 | 190 | 224 | 27,84 | 81,06 | 73,61 | 0,77 |
| **05** | **90** | 90 | 337 | 181 | 214 | 28,97 | 82,82 | 73,04 | 0,78 |
| **07** | **05** | 78 | 245 | 197 | 195 | 25,00 | 82,72 | 74,24 | 0,78 |
| **07** | **10** | 77 | 243 | 195 | 212 | 25,04 | 83,58 | 74,27 | 0,79 |
| **07** | **30** | 72 | 252 | 196 | 219 | 25,33 | 83,17 | 73,28 | 0,78 |
| **07** | **50** | 76 | 273 | 196 | 232 | 26,80 | 81,94 | 73,31 | 0,77 |
| **07** | **70** | 91 | 300 | 186 | 222 | 28,02 | 80,77 | 73,12 | 0,77 |
| **07** | **90** | 96 | 335 | 183 | 221 | 29,53 | 82,13 | 72,45 | 0,77 |
| **10** | **05** | 79 | 242 | 199 | 207 | 25,09 | 83,00 | 74,44 | 0,78 |
| **10** | **10** | 76 | 242 | 198 | 221 | 25,22 | 83,12 | 74,19 | 0,78 |
| **10** | **30** | 76 | 249 | 195 | 223 | 25,54 | 83,39 | 73,70 | 0,78 |
| **10** | **50** | 84 | 277 | 196 | 223 | 27,08 | 81,54 | 73,16 | 0,77 |
| **10** | **70** | 97 | 304 | 195 | 219 | 28,51 | 80,41 | 72,55 | 0,76 |
| **10** | **90** | 102 | 340 | 188 | 222 | 29,90 | 80,78 | 71,61 | 0,76 |
| **15** | **05** | 81 | 242 | 208 | 203 | 25,57 | 82,16 | 73,17 | 0,77 |
| **15** | **10** | 86 | 239 | 208 | 218 | 25,84 | 82,18 | 73,33 | 0,78 |
| **15** | **30** | 85 | 254 | 216 | 209 | 26,49 | 81,37 | 72,16 | 0,76 |
| **15** | **50** | 94 | 284 | 200 | 225 | 27,87 | 80,98 | 72,32 | 0,76 |
| **15** | **70** | 92 | 301 | 190 | 214 | 28,25 | 80,70 | 72,52 | 0,76 |
| **15** | **90** | 107 | 354 | 186 | 223 | 30,77 | 79,17 | 70,58 | 0,75 |
| **20** | **05** | 83 | 241 | 204 | 211 | 26,02 | 80,58 | 72,85 | 0,77 |
| **20** | **10** | 89 | 242 | 216 | 224 | 26,83 | 79,88 | 72,33 | 0,76 |
| **20** | **30** | 84 | 250 | 219 | 232 | 26,78 | 80,68 | 72,46 | 0,76 |
| **20** | **50** | 92 | 282 | 198 | 226 | 27,69 | 80,67 | 72,86 | 0,77 |
| **20** | **70** | 98 | 320 | 194 | 213 | 29,27 | 80,44 | 71,95 | 0,76 |
| **20** | **90** | 100 | 351 | 186 | 231 | 30,37 | 80,07 | 70,99 | 0,75 |
| **inf** | **inf** | 125 | 376 | 163 | 319 | 33,90 | 70,65 | 70,00 | 0,70 |

TABLE A.11 – Performances (SER), erreurs d'Insertion (I), de Délétion (D), de Type (T), d'Extension (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit+CasEN` selon la Fréquence (F) et la Confiance (C)

**Etape, approche** `Logit`

Les expériences avec de motifs classiques sur Etape donnent de moins bonnes performances qu'avec les motifs de segments. Comme pour Ester2, nous remarquons un fort taux d'erreurs de délétion.

| F | C | SER | Co | I | D | S | E | P | R | F |
|---|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 36.2 | 64.6 | 5.7 | 24.5 | 10.9 | 41.1 | 79.5 | 64.6 | 71.3 |
| **03** | **10** | 36.1 | 64.7 | 5.7 | 24.4 | 10.9 | 41.0 | 79.6 | 64.7 | 71.4 |
| **03** | **30** | 36.5 | 64.4 | 5.6 | 25.1 | 10.6 | 41.3 | 79.9 | 64.4 | 71.3 |
| **03** | **50** | 36.9 | 63.8 | 5.5 | 25.5 | 10.7 | 41.7 | 79.7 | 63.8 | 70.9 |
| **03** | **70** | 38.6 | 61.6 | 5.0 | 28.0 | 10.4 | 43.3 | 80.0 | 61.6 | 69.6 |
| **03** | **90** | 45.7 | 53.2 | 3.2 | 37.2 | 9.6 | 50.0 | 80.6 | 53.2 | 64.1 |
| **05** | **05** | 36.5 | 64.2 | 5.6 | 24.9 | 10.9 | 41.4 | 79.5 | 64.2 | 71.1 |
| **05** | **10** | 36.5 | 64.1 | 5.5 | 25.1 | 10.8 | 41.4 | 79.8 | 64.1 | 71.1 |
| **05** | **30** | 37.3 | 63.6 | 5.6 | 25.8 | 10.6 | 42.0 | 79.7 | 63.6 | 70.7 |
| **05** | **50** | 37.8 | 62.7 | 5.4 | 26.5 | 10.8 | 42.7 | 79.5 | 62.7 | 70.1 |
| **05** | **70** | 40.2 | 60.0 | 5.0 | 29.4 | 10.6 | 45.0 | 79.4 | 60.0 | 68.3 |
| **05** | **90** | 48.2 | 49.9 | 2.9 | 39.2 | 10.9 | 53.0 | 78.3 | 49.9 | 60.9 |
| **07** | **05** | 36.9 | 63.8 | 5.6 | 25.3 | 10.9 | 41.8 | 79.5 | 63.8 | 70.8 |
| **07** | **10** | 37.1 | 63.8 | 5.7 | 25.5 | 10.7 | 41.9 | 79.6 | 63.8 | 70.8 |
| **07** | **30** | 37.8 | 63.2 | 5.8 | 26.2 | 10.6 | 42.6 | 79.4 | 63.2 | 70.4 |
| **07** | **50** | 38.4 | 62.1 | 5.3 | 27.2 | 10.7 | 43.2 | 79.5 | 62.1 | 69.7 |
| **07** | **70** | 40.7 | 59.4 | 4.9 | 29.9 | 10.7 | 45.5 | 79.1 | 59.4 | 67.9 |
| **07** | **90** | 49.1 | 49.0 | 2.8 | 40.5 | 10.5 | 53.8 | 78.6 | 49.0 | 60.4 |
| **10** | **05** | 38.7 | 62.1 | 5.6 | 27.0 | 10.9 | 43.5 | 78.9 | 62.1 | 69.5 |
| **10** | **10** | 39.0 | 61.9 | 5.7 | 27.2 | 10.9 | 43.8 | 78.8 | 61.9 | 69.3 |
| **10** | **30** | 39.2 | 61.6 | 5.6 | 27.6 | 10.8 | 44.0 | 79.0 | 61.6 | 69.3 |
| **10** | **50** | 39.3 | 61.0 | 5.2 | 28.0 | 10.9 | 44.2 | 79.1 | 61.0 | 68.9 |
| **10** | **70** | 42.3 | 57.9 | 5.1 | 31.0 | 11.1 | 47.2 | 78.2 | 57.9 | 66.5 |
| **10** | **90** | 51.4 | 46.6 | 2.7 | 42.5 | 10.9 | 56.1 | 77.4 | 46.6 | 58.1 |
| **15** | **05** | 39.0 | 61.9 | 5.7 | 27.2 | 10.9 | 43.8 | 78.8 | 61.9 | 69.3 |
| **15** | **10** | 39.7 | 61.5 | 5.9 | 27.9 | 10.6 | 44.4 | 78.8 | 61.5 | 69.1 |
| **15** | **30** | 40.0 | 60.9 | 5.7 | 28.4 | 10.7 | 44.8 | 78.8 | 60.9 | 68.7 |
| **15** | **50** | 40.5 | 60.1 | 5.4 | 28.9 | 11.0 | 45.3 | 78.6 | 60.1 | 68.1 |
| **15** | **70** | 43.4 | 56.9 | 5.2 | 32.1 | 11.0 | 48.3 | 77.8 | 56.9 | 65.7 |
| **15** | **90** | 53.9 | 43.8 | 2.7 | 44.8 | 11.4 | 58.8 | 75.7 | 43.8 | 55.5 |
| **20** | **05** | 39.7 | 60.7 | 5.4 | 27.9 | 11.4 | 44.7 | 78.4 | 60.7 | 68.4 |
| **20** | **10** | 40.5 | 60.2 | 5.6 | 28.7 | 11.1 | 45.4 | 78.3 | 60.2 | 68.1 |
| **20** | **30** | 41.8 | 58.6 | 5.5 | 30.0 | 11.3 | 46.8 | 77.7 | 58.6 | 66.8 |
| **20** | **50** | 42.9 | 57.1 | 5.2 | 31.1 | 11.8 | 48.1 | 77.1 | 57.1 | 65.6 |
| **20** | **70** | 45.9 | 54.1 | 5.0 | 34.9 | 11.1 | 50.9 | 77.1 | 54.1 | 63.6 |
| **20** | **90** | 57.4 | 40.5 | 2.5 | 48.7 | 10.7 | 62.0 | 75.4 | 40.5 | 52.7 |
| **inf** | **inf** | 78.8 | 12.3 | 2.2 | 62.1 | 25.6 | 89.8 | 30.8 | 12.3 | 17.6 |

TABLE A.12 – Performances (SER), taux d'entités correctes (Co), Insérées (I), omises (D), substitutées (S), Erronées (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit` selon la Fréquence (F) et la Confiance (C)

**Etape, approche** `Logit+Segs`

Pour Etape, comme reporté en figure 9.7, les meilleures performances sont atteintes avec les motifs de segments. Nous voyons que le gain est réparti sur toutes les erreurs.

| F | C | SER | Co | I | D | S | E | P | R | F |
|---|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 35.9 | 64.9 | 5.6 | 24.2 | 10.8 | 40.7 | 79.8 | 64.9 | 71.6 |
| **03** | **10** | 35.9 | 64.8 | 5.6 | 24.4 | 10.7 | 40.7 | 79.9 | 64.8 | 71.6 |
| **03** | **30** | 36.2 | 64.7 | 5.6 | 24.6 | 10.7 | 40.9 | 79.8 | 64.7 | 71.5 |
| **03** | **50** | 36.4 | 64.2 | 5.6 | 24.8 | 11.0 | 41.4 | 79.5 | 64.2 | 71.0 |
| **03** | **70** | 37.7 | 62.6 | 5.0 | 26.9 | 10.5 | 42.4 | 80.2 | 62.6 | 70.3 |
| **03** | **90** | 44.8 | 53.6 | 3.2 | 35.7 | 10.7 | 49.6 | 79.5 | 53.6 | 64.0 |
| **05** | **05** | 36.1 | 64.5 | 5.5 | 24.7 | 10.8 | 41.0 | 79.8 | 64.5 | 71.3 |
| **05** | **10** | 36.4 | 64.4 | 5.6 | 24.8 | 10.8 | 41.2 | 79.8 | 64.4 | 71.3 |
| **05** | **30** | 36.6 | 64.2 | 5.5 | 25.0 | 10.8 | 41.4 | 79.7 | 64.2 | 71.1 |
| **05** | **50** | 36.9 | 63.5 | 5.3 | 25.7 | 10.8 | 41.8 | 79.8 | 63.5 | 70.7 |
| **05** | **70** | 38.7 | 61.4 | 4.7 | 28.1 | 10.5 | 43.4 | 80.1 | 61.4 | 69.5 |
| **05** | **90** | 47.4 | 50.6 | 3.0 | 38.2 | 11.2 | 52.5 | 78.0 | 50.6 | 61.4 |
| **07** | **05** | 36.8 | 64.1 | 5.7 | 25.1 | 10.8 | 41.7 | 79.5 | 64.1 | 70.9 |
| **07** | **10** | 36.5 | 64.2 | 5.7 | 24.8 | 11.0 | 41.4 | 79.4 | 64.2 | 71.0 |
| **07** | **30** | 36.8 | 64.0 | 5.7 | 24.9 | 11.1 | 41.8 | 79.1 | 64.0 | 70.7 |
| **07** | **50** | 37.5 | 62.9 | 5.3 | 26.2 | 10.9 | 42.5 | 79.4 | 62.9 | 70.2 |
| **07** | **70** | 39.4 | 60.8 | 5.0 | 28.5 | 10.7 | 44.2 | 79.5 | 60.8 | 68.9 |
| **07** | **90** | 48.5 | 49.4 | 2.8 | 39.6 | 11.0 | 53.4 | 78.1 | 49.4 | 60.5 |
| **10** | **05** | 37.6 | 63.1 | 5.6 | 25.9 | 11.0 | 42.5 | 79.2 | 63.1 | 70.2 |
| **10** | **10** | 37.7 | 63.0 | 5.7 | 25.9 | 11.1 | 42.7 | 79.0 | 63.0 | 70.1 |
| **10** | **30** | 38.1 | 62.6 | 5.5 | 26.5 | 10.9 | 43.0 | 79.2 | 62.6 | 69.9 |
| **10** | **50** | 38.3 | 62.2 | 5.4 | 27.0 | 10.8 | 43.1 | 79.4 | 62.2 | 69.8 |
| **10** | **70** | 40.5 | 59.9 | 5.2 | 29.6 | 10.5 | 45.3 | 79.2 | 59.9 | 68.2 |
| **10** | **90** | 50.2 | 47.6 | 2.6 | 41.5 | 11.0 | 55.1 | 77.8 | 47.6 | 59.0 |
| **15** | **05** | 38.2 | 62.5 | 5.6 | 26.4 | 11.2 | 43.1 | 78.9 | 62.5 | 69.7 |
| **15** | **10** | 38.6 | 62.1 | 5.6 | 26.6 | 11.2 | 43.5 | 78.6 | 62.1 | 69.4 |
| **15** | **30** | 39.3 | 61.3 | 5.5 | 27.8 | 10.9 | 44.2 | 78.9 | 61.3 | 69.0 |
| **15** | **50** | 40.2 | 60.4 | 5.5 | 28.6 | 10.9 | 45.1 | 78.6 | 60.4 | 68.3 |
| **15** | **70** | 42.0 | 58.7 | 5.5 | 30.6 | 10.6 | 46.7 | 78.5 | 58.7 | 67.2 |
| **15** | **90** | 52.6 | 44.6 | 2.5 | 43.6 | 11.8 | 57.9 | 75.7 | 44.6 | 56.2 |
| **20** | **05** | 38.9 | 61.6 | 5.4 | 27.2 | 11.2 | 43.8 | 78.8 | 61.6 | 69.2 |
| **20** | **10** | 39.3 | 61.2 | 5.4 | 27.8 | 11.1 | 44.2 | 78.8 | 61.2 | 68.9 |
| **20** | **30** | 41.2 | 59.5 | 5.6 | 29.5 | 11.0 | 46.1 | 78.2 | 59.5 | 67.6 |
| **20** | **50** | 41.8 | 58.7 | 5.4 | 30.4 | 11.0 | 46.7 | 78.2 | 58.7 | 67.0 |
| **20** | **70** | 43.9 | 56.5 | 5.3 | 32.8 | 10.7 | 48.7 | 78.0 | 56.5 | 65.6 |
| **20** | **90** | 56.4 | 41.0 | 2.4 | 47.8 | 11.2 | 61.3 | 75.2 | 41.0 | 53.1 |
| **inf** | **inf** | 78.8 | 12.3 | 2.2 | 62.1 | 25.6 | 89.8 | 30.8 | 12.3 | 17.6 |

TABLE A.13 – Performances (SER), taux d'entités correctes (Co), Insérées (I), omises (D), substitutées (S), Erronées (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit+Segs` selon la Fréquence (F) et la Confiance (C)

**Etape, approche** `Logit-Dicos`

Comme pour Ester2, nous notons ici la dégradation importante des performances lorsque les ressources lexicales sont désactivées.

| F | C | SER | Co | I | D | S | E | P | R | F |
|---|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 45.2 | 53.5 | 5.9 | 30.2 | 16.3 | 52.4 | 70.7 | 53.5 | 60.9 |
| **03** | **10** | 45.3 | 53.9 | 6.3 | 29.8 | 16.4 | 52.4 | 70.4 | 53.9 | 61.0 |
| **03** | **30** | 45.3 | 53.3 | 5.9 | 30.3 | 16.4 | 52.6 | 70.5 | 53.3 | 60.7 |
| **03** | **50** | 45.6 | 52.9 | 5.7 | 31.0 | 16.2 | 52.8 | 70.8 | 52.9 | 60.5 |
| **03** | **70** | 51.1 | 46.9 | 4.9 | 37.3 | 15.8 | 58.0 | 69.3 | 46.9 | 56.0 |
| **05** | **05** | 45.7 | 53.0 | 5.8 | 31.1 | 15.9 | 52.7 | 71.0 | 53.0 | 60.7 |
| **05** | **10** | 45.9 | 53.0 | 6.0 | 30.8 | 16.1 | 53.0 | 70.5 | 53.0 | 60.5 |
| **05** | **30** | 46.3 | 52.5 | 5.9 | 31.6 | 15.9 | 53.4 | 70.7 | 52.5 | 60.3 |
| **05** | **50** | 46.5 | 52.0 | 5.5 | 32.4 | 15.6 | 53.5 | 71.2 | 52.0 | 60.1 |
| **05** | **70** | 53.2 | 44.2 | 4.6 | 39.4 | 16.4 | 60.4 | 67.8 | 44.2 | 53.5 |
| **07** | **05** | 46.2 | 52.5 | 5.7 | 31.6 | 15.9 | 53.2 | 70.9 | 52.5 | 60.3 |
| **07** | **10** | 46.0 | 52.5 | 5.5 | 31.5 | 16.0 | 53.0 | 70.9 | 52.5 | 60.3 |
| **07** | **30** | 46.5 | 52.0 | 5.6 | 32.0 | 16.0 | 53.6 | 70.6 | 52.0 | 59.9 |
| **07** | **50** | 47.2 | 51.2 | 5.4 | 33.3 | 15.4 | 54.1 | 71.1 | 51.2 | 59.6 |
| **07** | **70** | 54.8 | 42.5 | 4.6 | 40.7 | 16.8 | 62.1 | 66.5 | 42.5 | 51.8 |
| **10** | **05** | 46.3 | 52.1 | 5.4 | 32.1 | 15.8 | 53.3 | 71.0 | 52.1 | 60.1 |
| **10** | **10** | 46.6 | 51.9 | 5.5 | 32.4 | 15.7 | 53.6 | 71.0 | 51.9 | 60.0 |
| **10** | **30** | 47.2 | 51.1 | 5.1 | 33.5 | 15.4 | 54.1 | 71.3 | 51.1 | 59.5 |
| **10** | **50** | 48.3 | 50.1 | 5.1 | 34.9 | 15.0 | 55.0 | 71.4 | 50.1 | 58.9 |
| **10** | **70** | 56.7 | 40.0 | 4.0 | 43.0 | 17.0 | 64.0 | 65.5 | 40.0 | 49.7 |
| **15** | **05** | 48.6 | 49.8 | 5.4 | 34.4 | 15.8 | 55.6 | 70.1 | 49.8 | 58.2 |
| **15** | **10** | 48.8 | 49.8 | 5.5 | 34.7 | 15.6 | 55.8 | 70.2 | 49.8 | 58.2 |
| **15** | **30** | 50.0 | 48.8 | 5.6 | 35.7 | 15.5 | 56.9 | 69.7 | 48.8 | 57.4 |
| **15** | **50** | 51.0 | 47.2 | 5.1 | 37.5 | 15.3 | 57.8 | 69.8 | 47.2 | 56.4 |
| **15** | **70** | 59.9 | 37.2 | 3.9 | 46.9 | 15.9 | 66.7 | 65.3 | 37.2 | 47.4 |
| **20** | **05** | 50.4 | 48.2 | 5.6 | 36.1 | 15.7 | 57.3 | 69.4 | 48.2 | 56.9 |
| **20** | **10** | 51.1 | 47.8 | 5.8 | 36.4 | 15.8 | 58.0 | 68.9 | 47.8 | 56.4 |
| **20** | **30** | 52.5 | 46.3 | 5.6 | 38.2 | 15.5 | 59.3 | 68.7 | 46.3 | 55.3 |
| **20** | **50** | 52.9 | 45.5 | 5.2 | 39.4 | 15.1 | 59.7 | 69.1 | 45.5 | 54.9 |
| **20** | **70** | 61.4 | 35.0 | 3.7 | 48.0 | 17.0 | 68.7 | 62.9 | 35.0 | 45.0 |
| **inf** | **inf** | 82.2 | 10.2 | 1.5 | 68.4 | 21.5 | 91.3 | 30.7 | 10.2 | 15.3 |

TABLE A.14 – Performances (SER), taux d'entités correctes (Co), Insérées (I), omises (D), substitutées (S), Erronées (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit-Dicos` selon la Fréquence (F) et la Confiance (C)

**Etape, approche** `Logit+Test`

L'introduction des donnnées d'évaluation pour extraire les règles et paramétrer le système améliore ici aussi les résultats, mais dans une moindre mesure.

| F | C | SER | Co | I | D | S | E | P | R | F |
|---|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 26.3 | 73.3 | 3.2 | 18.6 | 8.1 | 29.9 | 86.6 | 73.3 | 79.4 |
| **03** | **10** | 26.4 | 73.2 | 3.2 | 18.7 | 8.2 | 30.1 | 86.5 | 73.2 | 79.3 |
| **03** | **30** | 27.4 | 72.4 | 3.5 | 19.3 | 8.3 | 31.1 | 86.0 | 72.4 | 78.6 |
| **03** | **50** | 28.1 | 71.7 | 3.6 | 20.0 | 8.3 | 31.9 | 85.8 | 71.7 | 78.1 |
| **03** | **70** | 29.9 | 69.5 | 3.2 | 22.2 | 8.3 | 33.7 | 85.8 | 69.5 | 76.8 |
| **03** | **90** | 38.7 | 59.1 | 1.9 | 31.9 | 9.1 | 42.8 | 84.4 | 59.1 | 69.5 |
| **05** | **05** | 27.9 | 72.0 | 3.7 | 19.6 | 8.4 | 31.7 | 85.6 | 72.0 | 78.2 |
| **05** | **10** | 28.2 | 71.9 | 3.9 | 19.7 | 8.5 | 32.0 | 85.4 | 71.9 | 78.0 |
| **05** | **30** | 29.0 | 70.9 | 3.8 | 20.5 | 8.6 | 32.9 | 85.1 | 70.9 | 77.3 |
| **05** | **50** | 29.7 | 70.0 | 3.7 | 21.3 | 8.7 | 33.7 | 84.9 | 70.0 | 76.8 |
| **05** | **70** | 31.5 | 67.9 | 3.4 | 23.3 | 8.8 | 35.5 | 84.8 | 67.9 | 75.4 |
| **05** | **90** | 41.7 | 56.0 | 1.8 | 34.8 | 9.2 | 45.9 | 83.5 | 56.0 | 67.0 |
| **07** | **05** | 29.8 | 70.4 | 4.1 | 21.0 | 8.6 | 33.7 | 84.8 | 70.4 | 76.9 |
| **07** | **10** | 29.9 | 70.3 | 4.1 | 21.0 | 8.7 | 33.8 | 84.6 | 70.3 | 76.8 |
| **07** | **30** | 30.7 | 69.4 | 4.0 | 21.8 | 8.8 | 34.6 | 84.4 | 69.4 | 76.1 |
| **07** | **50** | 31.4 | 68.5 | 3.9 | 22.6 | 8.9 | 35.4 | 84.2 | 68.5 | 75.6 |
| **07** | **70** | 33.3 | 66.5 | 3.7 | 24.8 | 8.7 | 37.2 | 84.3 | 66.5 | 74.4 |
| **07** | **90** | 43.9 | 53.4 | 1.9 | 36.5 | 10.1 | 48.5 | 81.6 | 53.4 | 64.6 |
| **10** | **05** | 31.1 | 69.2 | 4.3 | 22.0 | 8.8 | 35.1 | 84.1 | 69.2 | 75.9 |
| **10** | **10** | 31.2 | 69.0 | 4.3 | 21.9 | 9.1 | 35.3 | 83.7 | 69.0 | 75.7 |
| **10** | **30** | 32.2 | 67.9 | 4.2 | 23.1 | 9.1 | 36.3 | 83.7 | 67.9 | 75.0 |
| **10** | **50** | 32.8 | 67.2 | 4.1 | 23.6 | 9.2 | 36.9 | 83.4 | 67.2 | 74.4 |
| **10** | **70** | 34.7 | 65.1 | 3.9 | 25.8 | 9.2 | 38.8 | 83.3 | 65.1 | 73.1 |
| **10** | **90** | 46.6 | 50.4 | 1.8 | 38.8 | 10.7 | 51.3 | 80.1 | 50.4 | 61.9 |
| **15** | **05** | 32.6 | 67.5 | 4.3 | 23.1 | 9.3 | 36.8 | 83.2 | 67.5 | 74.6 |
| **15** | **10** | 32.9 | 67.1 | 4.3 | 23.4 | 9.4 | 37.1 | 83.0 | 67.1 | 74.2 |
| **15** | **30** | 34.7 | 65.8 | 4.7 | 24.9 | 9.3 | 38.9 | 82.5 | 65.8 | 73.2 |
| **15** | **50** | 34.8 | 65.6 | 4.5 | 25.2 | 9.2 | 39.0 | 82.7 | 65.6 | 73.1 |
| **15** | **70** | 36.6 | 63.5 | 4.3 | 27.2 | 9.3 | 40.8 | 82.4 | 63.5 | 71.7 |
| **15** | **90** | 50.1 | 47.3 | 2.0 | 42.4 | 10.3 | 54.7 | 79.4 | 47.3 | 59.3 |
| **20** | **05** | 34.1 | 65.9 | 4.3 | 24.2 | 9.9 | 38.4 | 82.3 | 65.9 | 73.2 |
| **20** | **10** | 34.5 | 65.4 | 4.3 | 24.6 | 9.9 | 38.8 | 82.2 | 65.4 | 72.9 |
| **20** | **30** | 36.3 | 64.0 | 4.6 | 26.5 | 9.6 | 40.6 | 81.9 | 64.0 | 71.8 |
| **20** | **50** | 37.1 | 63.0 | 4.4 | 27.4 | 9.7 | 41.4 | 81.7 | 63.0 | 71.1 |
| **20** | **70** | 39.2 | 60.8 | 4.2 | 29.9 | 9.4 | 43.4 | 81.8 | 60.8 | 69.7 |
| **20** | **90** | 53.6 | 43.7 | 1.9 | 45.9 | 10.4 | 58.2 | 78.0 | 43.7 | 56.0 |
| **inf** | **inf** | 74.4 | 17.5 | 1.3 | 60.9 | 21.6 | 83.8 | 43.4 | 17.5 | 24.9 |

TABLE A.15 – Performances (SER), taux d'entités correctes (Co), Insérées (I), omises (D), substitutées (S), Erronées (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit+Test` selon la Fréquence (F) et la Confiance (C)

**Etape, approches `Logit-D25`, `Logit-D50` et `Logit-D75`**

Comme pour Ester2, filtrer les règles extraites pénalise relativement peu le système dans un premier temps.

| F | C | SER | Co | I | D | S | E | P | R | F |
|---|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 36.2 | 64.6 | 5.6 | 24.6 | 10.9 | 41.1 | 79.7 | 64.6 | 71.3 |

TABLE A.16 – Performances (SER), taux d'entités correctes (Co), Insérées (I), omises (D), substitutées (S), Erronées (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit-D25` selon la Fréquence (F) et la Confiance (C)

| F | C | SER | Co | I | D | S | E | P | R | F |
|---|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 36.7 | 63.8 | 5.4 | 25.2 | 11.0 | 41.6 | 79.5 | 63.8 | 70.8 |

TABLE A.17 – Performances (SER), taux d'entités correctes (Co), Insérées (I), omises (D), substitutées (S), Erronées (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit-D50` selon la Fréquence (F) et la Confiance (C)

| F | C | SER | Co | I | D | S | E | P | R | F |
|---|---|---|---|---|---|---|---|---|---|---|
| **03** | **05** | 39.0 | 61.3 | 5.4 | 27.0 | 11.7 | 44.1 | 78.2 | 61.3 | 68.7 |

TABLE A.18 – Performances (SER), taux d'entités correctes (Co), Insérées (I), omises (D), substitutées (S), Erronées (E), Précision (P), Rappel (R), F-mesure (Fm) pour l'approche `Logit-D75` selon la Fréquence (F) et la Confiance (C)

# Annexe B

# Extraits d'annotation

### Annotation Ester2

```
20080118_1000_1100_inter 1 Sonia_Devillers 536.227 538.970 <o,f0,female> moi je voulais juste
dire , ils ont , ils ont un , quand même un seul point commun ,
20080118_1000_1100_inter 1 Sonia_Devillers 538.970 542.282 <o,f0,female> [r] c'est une euh euh
finalement euh
20080118_1000_1100_inter 1 Sonia_Devillers 542.282 546.649 <o,f0,female> une certitude [r] que
la politique ne pourra plus répondre à leurs attentes
20080118_1000_1100_inter 1 Sonia_Devillers 546.649 549.762 <o,f0,female> [r] et que finalement
euh on appelait les
20080118_1000_1100_inter 1 Sonia_Devillers 549.762 554.191 <o,f0,female> [r] les euh , comment
dire , les démarches et les logiques politiques traditionnelles , c'est-à-dire meetings ,
20080118_1000_1100_inter 1 Sonia_Devillers 554.191 555.767 <o,f0,female> pétitions , entrer
dans un parti politique ,
20080118_1000_1100_inter 1 Sonia_Devillers 555.767 557.336 <o,f0,female> [r] ne servent plus
à rien .
20080118_1000_1100_inter 1 Sonia_Devillers 557.336 559.484 <o,f0,female> [r] et que par conséquent
face à cette euh
20080118_1000_1100_inter 1 Sonia_Devillers 559.484 562.496 <o,f0,female> [r] à ce vide politique
[r] mais aussi à l'apathie ,
20080118_1000_1100_inter 1 Sonia_Devillers 562.496 564.766 <o,f0,female> [r] ce qu'ils ressentent
comme l'apathie des gens ,
20080118_1000_1100_inter 1 Sonia_Devillers 564.766 568.449 <o,f0,female> [r] la seule manière
, c'est euh d'entrer en scène soi-même .
20080118_1000_1100_inter 1 excluded_region 568.449 569.344 <o,,unknown> ignore_time_segment_in_scoring
20080118_1000_1100_inter 1 Caroline_Glorion 569.344 571.200 <o,f0,female> ça justement précisément
ma question que
20080118_1000_1100_inter 1 Caroline_Glorion 571.200 573.480 <o,f0,female> [r] qu'est pas très
clair dans l'article et en tout cas
20080118_1000_1100_inter 1 Caroline_Glorion 573.480 574.737 <o,f0,female> [b] assez peu abordé
20080118_1000_1100_inter 1 Caroline_Glorion 574.737 577.432 <o,f0,female> [r] c'est l(e) , cette
, ce rapport au politique hein ,
20080118_1000_1100_inter 1 Caroline_Glorion 577.432 580.649 <o,f0,female> [r] parce que vous
dites à la fois , ce sont pas des individus qui réagissent euh
20080118_1000_1100_inter 1 Caroline_Glorion 580.649 583.681 <o,f0,female> [r] uniquement pour
eux-mêmes , ils essaient de faire des actions collectives
20080118_1000_1100_inter 1 Caroline_Glorion 583.681 586.861 <o,f0,female> [r] et jusqu'à preuve
du contraire c'est quand même au sein des partis politiques ,
```

### Annotation Etape

```
donc le premier thème ce sera la mort dans les <loc.adm.reg> <kind> Pays </kind> de <name> Savoie
</name> </loc.adm.reg> de l' <time.date.abs> <reference-era> Antiquité </reference-era> </time.date.abs>
à nos jours .
et puis
euh
un certain nombre de manifestations autour : un   salon du livre  d'  histoire  des <loc.adm.reg>
<kind> Pays </kind> de <name> Savoie </name> </loc.adm.reg>  ,
des manifestations culturelles , des documentaires ,
euh
des
une <org.ent> <kind> association </kind> d' <name> escrime </name> </org.ent> qui va intervenir
pour nous faire des démonstrations
euh
pour le premier festival .
<pers.ind> <name.first> Guillaume </name.first> </pers.ind> on va rappeler une chose : premier
rendez-vous <time.date.abs> <month> juin </month> <year> 2011 </year> </time.date.abs> , c'
est ça ?
on a les dates déjà ?
alors ça commencera <time.date.abs> <time-modifier> le </time-modifier> <week> vendredi </week>
<day> 24 </day> <month> juin </month> <year> 2011 </year> </time.date.abs> <time.hour.rel> <time-modifier>
en fin d' </time-modifier> <name> après-midi </name> </time.hour.rel> pour
le
l' inauguration officielle
et <time.date.abs> <time-modifier> jusqu' au </time-modifier> <week> dimanche </week> <time-modifier>
après-midi </time-modifier> </time.date.abs> , <time.date.abs> <day> 26 </day> <month> juin
</month> </time.date.abs> .
alors on va rappeler que
ce
ce   festival   sera ouvert au grand public . ça veut dire que tout le monde , tous ceux qui
s' intéressent à l' histoire auront leur place dans ce   festival   .
alors c' est vraiment l' objectif . c' est-à-dire que c' est un festival qui va être
de très bon niveau . c' est-à-dire contrôlé par des <func.coll> <kind> universitaires </kind>
</func.coll> , des <func.coll> <kind> historiens </kind> </func.coll> ,
des passionnés .
mais le but c' est vraiment de l' ouvrir au public le plus large , à tous les <pers.coll> <demonym>
savoyards </demonym> </pers.coll> , et au-delà .
euh
tous les <pers.coll> <kind> passionnés </kind> <qualifier> jeunes </qualifier> </pers.coll>
, plus âgés , etc . donc vraiment d' en faire une fête de l' histoire ouverte au plus grand
nombre .
on a quand même l' impression que <loc.adm.town><name> La Roche </name></loc.adm.town> ,
certes est connue , [...]
```

# Annexe C

# Journée ATALA : Reconnaissance d'Entités Nommées - Nouvelles Frontières & Nouvelles Approches

Journée de l'ATALA : Reconnaissance d'Entités Nommées, Nouvelles Frontières et Nouvelles Approches
Date : lundi 20 juin 2011
Lieu : Centre Malesherbes - Amphi 122, 108 boulevard Malesherbes, 75017 Paris
Site web : http://tln.li.univ-tours.fr/Tln_Colloques/Tln_REN2011.html

Programme
---
9h30 :Accueil
10h00 : Méthodologie et ressources pour la résolution d'entités nommées dans des dépêches d'agence
R. Stern, B. Sagot - Alpage, INRIA Paris-Rocquencourt & Université Paris 7, Agence France-Presse - Medialab
10h30 : Caractérisation des entités nommées du point de vue du rapport élément/collection. Le cas des "noms propres collectifs"
M. Lecolle - Université Paul Verlaine-Metz, CELTED
11h00 : Pause café
11h30 : Les entités nommées dans le programme QUAERO
S. Rosset , C. Grouin , O. Galibert, P. Zweigenbaum , K. Fort, L. Quintard - LIMSI-CNRS, LNE, INIST-CNRS, LIPN
12h00 : Discussion : Corpus (francophones) annotés en Entités Nommées : bilan et perspectives
12h30 : Pause déjeuner
14h00 : Session posters :
* Reconnaissance des Entités Nommées par apprentissage automatique avec un corpus d'apprentissage bruité
F. Machen, I. Tellier - LIFO - Université d'Orléans, PASS Technologie
* Entités nommées et documents manuscrits en-ligne : comparaison avec l'oral pour des applications multimodales
S. Quiniou, C. Jacquin, E. Morin - LINA, Nantes
* RefGen : un système d'identification automatique de chaînes de référence
L. Longo, A. Todirascu - UR LiLPa - Université de Strasbourg
* Détection des entités nommées : un système symbolique open source à la disposition de la communauté
N. Friburger, D. Maurel - LI, Université François Rabelais Tours
* Reconnaissance d'Entités Nommées par extraction automatique de transducteurs
D. Nouvel, J.Y. Antoine, N. Friburger, A. Soulet - LI, Université François Rabelais Tours
15h30 : Extraction d'entités nommées à partir de graphes de mots

F. Bechet - Aix Marseille Université, LIF-CNRS, Marseille, France
16h00 : Vers une extraction automatique des événements dans les textes
B. Arnulphy, X. Tannier, A. Vilnat - Univ. Paris-Sud 11, LIMSI-CNRS
16h30 : Désambiguïsation des Entités Nommées : une approche basée sur l'extraction d'information
J.L. Bouraoui, P. Watrin - CENTAL, Université Catholique de Louvain
17h00 : Clôture de la journée

Thème de la journée
---
Le traitement des entités nommées a été un champ de recherche très actif ces dernières années. Si de nombreuses avancées ont été réalisées, on constate cependant que les technologies ont essentiellement évolué à la demande, à travers campagnes d'évaluation et besoins applicatifs. Aujourd'hui, diverses questions se posent, autant sur l'état des connaissances sur le sujet que du point de vue des perspectives de recherche.
D'un point de vue linguistique, ces unités linguistiques ne semblent pas encore complètement formalisées. Si de récents travaux en ont proposé plusieurs définitions, il reste difficile d'obtenir un consensus sur le sujet. D'où l'opportunité d'un échange, d'une part sur la nature des entités nommées, d'autre part (et rétroactivement) sur les normes, les typologies, les méthodes à utiliser dans un processus d'annotation (manuelle ou automatique). Ce sujets sont d'autant plus importants qu'à l'échelle française et européenne, des projets en cours de réalisation (ETAPE, QUAERO, ...) sont fortement dépendants de ces questions et tentent d'y apporter des réponses.
Par ailleurs, si la reconnaissance d'entités nommées a atteint une certaine maturité pour traiter, notamment, des textes journalistiques généralistes, de nombreuses questions restent ouvertes : intervention en sortie de reconnaissance automatique de la parole ou sur des textes récupérés en temps réel dans des flux d'actualités (dont entités hors-vocabulaire), suivi d'entités nommées (chaînes coréférentielles), désambiguisation hors-contexte / en contexte...

Comité de lecture
---
Jean-Yves Antoine (LI, Université François Rabelais Tours)
Frédéric Béchet (LIF, Université de la Méditerranée)
Thierry Charnois (GREYC, Université de Caen)
Bruno Crémilleux (GREYC, Université de Caen)
Maud Ehrmann (JRC, European Commission)
Nathalie Friburger (LI, Université François Rabelais Tours)
Christine Jacquin (LINA, Université de Nantes)
Denis Maurel (LI, Université François Rabelais Tours)
Emmanuel Morin (LINA, Université de Nantes)
Damien Nouvel (LI, Université François Rabelais Tours)
Thierry Poibeau (LaTTiCe, CNRS)
Sophie Rosset (LIMSI, CNRS)
Benoît Sagot (Alpage, INRIA)

Comité d'organisation
---
Jean-Yves Antoine (LI, Université François Rabelais Tours)
Nathalie Friburger (LI, Université François Rabelais Tours)
Denis Maurel (LI, Université François Rabelais Tours)
Damien Nouvel (LI, Université François Rabelais Tours)

# Bibliographie


[Min, 1993] (1993). *Mining Association Rules between Sets of Items in Large Databases.*

[Agrawal et Srikant, 1995] Agrawal, R. et Srikant, R. (1995). Mining sequential patterns. *In International Conference on Data Engineering (ICDE'95)*, pages 3–14.

[Berger *et al.*, 1996] Berger, A. L., Pietra, S. A. D. et Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71.

[Besançon *et al.*, 2006] Besançon, R., Embarek, M. et Ferret, O. (2006). Finding answers in the oedipe system by extracting and applying linguistic patterns. *In Conference of the Cross-Language Evaluation Forum (CLEF'06)*, pages 395–404.

[Bikel *et al.*, 1999] Bikel, D., Schwartz, R. et Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Machine Learning*, 34:211–231.

[Bonchi et Lucchese, 2005] Bonchi, F. et Lucchese, C. (2005). Pushing tougher constraints in frequent pattern mining. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, pages 114–124.

[Borthwick *et al.*, 1998] Borthwick, A., Sterling, J., Agichtein, E. et Grishman, R. (1998). Exploiting diverse knowledge sources via maximum entropy in named entity recognition. *In Sixth Workshop on Very Large Corpora (VLC'98)*, pages 152–160.

[Bouchou et Maurel, 2008] Bouchou, B. et Maurel, D. (2008). Prolexbase et lmf : vers un standard pour les ressources lexicales sur les noms propres. *Traitement Automatique des Langues (TAL)*, 49:61–88.

[Brun *et al.*, 2009a] Brun, C., Dessaigne, N., Ehrmann, M., Gaillard, B., Guillemin-Lanne, S., Jacquet, G., Kaplan, A., Kucharski, M., Martineau, C., Migeotte, A., Nakamura, T. et Voyatzi, S. (2009a). Une expérience de fusion pour l'annotation d'entités nommées. *In Traitement Automatique des Langues Naturelles (TALN'09).*

[Brun et Ehrmann, 2010] Brun, C. et Ehrmann, M. (2010). Un système de détection d'entités nommées adapté pour la campagne d'évaluation ester 2. *In Traitement Automatique du Langage Naturel (TALN'10).*

[Brun *et al.*, 2009b] Brun, C., Ehrmann, M. et Jacquet, G. (2009b). Résolution de métonymie des entités nommées : proposition d'une méthode hybride. *Traitement Automatique des Langues (TAL)*, 50:87–110.

[Brun et Hagège, 2008] Brun, C. et Hagège, C. (2008). Vérification sémantique pour l'annotation d'entités nommées. *In Traitement Automatique des Langues Naturelles (TALN'08)*.

[Budi et Bressan, 2007] Budi, I. et Bressan, S. (2007). Application of association rules mining to named entity recognition and co-reference resolution for the indonesian language. 2:426–446.

[Bunescu et Pasca, 2006] Bunescu, R. C. et Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. *In Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*.

[Béchet *et al.*, 2011] Béchet, F., Sagot, B. et Stern, R. (2011). Coopération de méthodes statistiques et symboliques pour l'adaptation non-supervisée d'un système d'étiquetage en entités nommées. *In Traitement Automatique des Langues Naturelles (TALN'11)*.

[Béchet et Roche, 2010] Béchet, N. et Roche, M. (2010). How to expand dictionaries with web-mining techniques, cogalex workshop. *In International Conference on Computational Linguistics (COLING'10)*, pages 33–37, Beijing, China.

[Califf et Mooney, 1999] Califf, M. E. et Mooney, R. J. (1999). Relational learning of pattern-match rules for information extraction. *In National Conference on Artificial Intelligence (AAAI'99)*, pages 328–334.

[Cellier et Charnois, 2010] Cellier, P. et Charnois, T. (2010). Fouille de données séquentielles d'itemsets pour l'apprentissage de patrons linguistiques. *In Traitement Automatique des Langues Naturelles (TALN'10)*.

[Charnois *et al.*, 2009] Charnois, T., Plantevit, M., Rigotti, C. et Crémilleux, B. (2009). Fouille de données séquentielles pour l'extraction d'information dans les textes. *Traitement Automatique des Langues (TAL)*, 50:87–110.

[Charton, 2009] Charton, E. (2009). Combinaison de contenus encyclopédiques multilingues pour une reconnaissance d'entités nommées en contexte. *In Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL'09)*.

[Charton *et al.*, 2011] Charton, E., Gagnon, M. et Ozell, B. (2011). Génération automatique de motifs de détection d'entités nommées en utilisant des contenus encyclopédiques. *In Traitement Automatique des Langues Naturelles (TALN'11)*.

[Charton et Torres-Moreno, 2009] Charton, E. et Torres-Moreno, J. M. (2009). Classification d'un contenu encyclopédique en vue d'un étiquetage par entités nommées. *In Traitement Automatique des Langues Naturelles (TALN'09)*.

[Chomsky, 1956] Chomsky, N. (1956). hree models for the description of language. *IRE Transactions on Information Theory*, 2:113–124.

[Chomsky, 1957] Chomsky, N. (1957). *Syntactic Structures*. Mouton.

[Col *et al.*, 2010] Col, G., Aptekman, J., Girault, S. et Victorri, B. (2010). Compositionnalité gestaltiste et construction du sens par instructions dynamiques. *CogniTextes*, 5.

[de Saussure, 1916] de Saussure, F. (1916). *Cours de linguistique générale*.

[Dinarelli et Rosset, 2011] Dinarelli, M. et Rosset, S. (2011). Models cascade for tree-structured named entity detection. *In International Joint Conference on Natural Language Processing (IJCNLP'11)*.

[Downey *et al.*, 2007] Downey, D., Broadhead, M. et Etzioni, O. (2007). Locating complex named entities in web text. *In International Joint Conference on Artifical Intelligence (IJCAI'07)*, pages 2733–2739.

[Dredze *et al.*, 2010] Dredze, M., McNamee, P., Rao, D., Gerber, A. et Finin, T. (2010). Entity disambiguation for knowledge base population. *In International Conference on Computational Linguistics (COLING'10)*, pages 277–285, Beijing, China.

[Ehrmann, 2008] Ehrmann, M. (2008). *Les entités nommées, de la linguistique au TAL : statut théorique et méthodes de désambiguïsation.* Thèse de doctorat, Université Paris VII, France.

[Ekbala *et al.*, 2010] Ekbala, A., Sourjikova, E., Frank, A. et Ponzetto, S. P. (2010). Assessing the challenge of fine-grained named entity recognition and classification. *In Annual Meeting of the Association for Computational Linguistics (ACL'10) - Named Entities Workshop*, pages 93–101, Uppsala, Sweden.

[Etzioni *et al.*, 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S. et Yates, A. (2005). Unsupervised named-entity extraction from the web : An experimental study. *Artificial Intelligence*, 165:91–134.

[Ezzat, 2010] Ezzat, M. (2010). Acquisition de grammaires locales pour l'extraction de relations entre entités nommées. *In Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL'10)*.

[Favre *et al.*, 2005] Favre, B., Béchet, F. et Nocera, P. (2005). Robust named entity extraction from large spoken archives. *In Joint Conference on Human Language Technology Conference and Empirical Methods in Natural Language Processing (HLT/EMNLP'05)*.

[Finkel et Manning, 2005] Finkel, J. R. et Manning, C. D. (2005). Nested named entity recognition. *In Conference on Empirical Methods in Natural Language Processing (EMNLP'09)*.

[Fischer *et al.*, 2005] Fischer, J., Heun, V. et Kramer, S. (2005). Fast frequent string mining using suffix arrays. *In 5th IEEE International Conference on Data Mining (ICDM'05)*, pages 609–612.

[Fort *et al.*, 2009] Fort, K., Ehrmann, M. et Nazarenko, A. (2009). Vers une méthodologie d'annotation des entités nommées en corpus ? *In Traitement Automatique des Langues Naturelles (TALN'09)*.

[Frege, 1892] Frege, G. (1892). *Über Sinn und Bedeutung.*

[Freitag et Kushmerick, 2000] Freitag, D. et Kushmerick, N. (2000). Boosted wrapper induction. *In European Conference on Artificial Intelligence (ECAI'00) - Workshop on Machine Learning for Information Extraction*, Berlin, Germany.

[Friburger, 2002] Friburger, N. (2002). *Reconnaissance automatique des noms propres : application à la classification automatique de textes journalistiques.* Thèse de doctorat, Université François-Rabelais Tours, France.

[Friburger, 2006] Friburger, N. (2006). Linguistique et reconnaissance automatique des noms propres. *Meta : Translators' Journal*, 51-4:637–650.

[Friburger et Maurel, 2004] Friburger, N. et Maurel, D. (2004). Finite-state transducer cascades to extract named entities in texts. *Theoretical Computer Sciences (TCS)*, 313: 93–104.

[Friburger et Maurel, 2011] Friburger, N. et Maurel, D. (2011). Writing and ordering fs cascades for nlp tasks using unitex.

[Galibert *et al.*, 2011] Galibert, O., Rosset, S., Grouin, C., Zweigenbaum, P. et Quintard, L. (2011). Structured and extended named entity evaluation in automatic speech transcriptions. *In International Joint Conference on Natural Language Processing (IJCNLP'11)*.

[Galliano *et al.*, 2009] Galliano, S., Gravier, G. et Chaubard, L. (2009). The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. *In Conference of the International Speech Communication Association (INTERSPEECH'09)*.

[Girault, 2008] Girault, T. (2008). Concept lattice mining for unsupervised named entity annotation. *In Concept Lattices and Their Applications (CLA'08)*.

[Grishman et Sundheim, 1996] Grishman, R. et Sundheim, B. (1996). Message undersrtanding conference - 6 : A brief history. *In International Conference on Computational Linguistics (COLING'96)*, pages 466–471, Copenhagen, Denmark.

[Grouin *et al.*, 2011] Grouin, C., Rosset, S., Zweigenbaum, P., Fort, K., Galibert, O. et Quintard, L. (2011). Proposal for an extension of traditional named entities : From guidelines to evaluation, an overview. pages 92–100.

[Han et Kamber, 2005] Han, J. et Kamber, M. (2005). *Data Mining : Concepts and Techniques*. Morgan Kaufmann Publishers Inc.

[Hingston, 2002] Hingston, P. (2002). Using finite state automata for sequence mining. *In Australasian Computer Science Conference (ACSC'02)*, pages 105–110.

[Isozaki et Kazawa, 2002] Isozaki, H. et Kazawa, H. (2002). Efficient support vector classifiers for named entity recognition. *In Conference on Computational linguistics (COLING'02)*.

[Kripke, 1972] Kripke, S. A. (1972). *Naming and Necessity*.

[Kushmerick *et al.*, 1997] Kushmerick, N., Weld, D. S. et Doorenbos, R. (1997). Wrapper induction for information extraction. *In International Joint Conference on Artificial Intelligence (IJCAI'97)*.

[Lafferty *et al.*, 2001] Lafferty, J., McCallum, A. et Pereira, F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *In International Conference on Machine Learning (ICML'01)*, pages 282–289.

[Lavergne *et al.*, 2010] Lavergne, T., Cappé, O. et Yvon, F. (2010). Practical very large scale crfs. *In Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 504–513.

[Lin *et al.*, 2010] Lin, B., Shah, R., Frederking, R. et Gershman, A. (2010). Cone : Metrics for automatic evaluation of named entity co-reference resolution. *In International Conference on Computational Linguistics (COLING'10)*, pages 931–939, Beijing, China.

[Liu *et al.*, 1998] Liu, B., Hsu, W. et Ma, Y. (1998). Integrating classification and association rule mining. *In ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD'98)*, pages 80–86.

[Makhoul *et al.*, 1999] Makhoul, J., Kubala, F., Schwartz, R. et Weischedel, R. (1999). Performance measures for information extraction. *In DARPA Broadcast News Workshop*, pages 249–252.

[Markert et Hahn, 2002] Markert, K. et Hahn, U. (2002). Understanding metonymies in discourse. *Artificial Intelligence (AI)*, 135:145–198.

[Maurel *et al.*, 2011] Maurel, D., Friburger, N., Nouvel, D., Eshkol-Taravella, I. et Antoine, J.-Y. (2011). Cascade de transducteurs : Applications autour des entités nommées. *Traitement Automatique des Langues (TAL)*, 52-1.

[McCallum *et al.*, 2000] McCallum, A., Freitag, D. et Pereira, F. (2000). Maximum entropy markov models for information extraction and segmentation. *In International Conference on Machine Learning (ICML'00)*, pages 591–598.

[McCallum et Li, 2003] McCallum, A. et Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *In Conference on Natural Language Learning (CONLL'03)*, pages 188–191.

[McDonald, 1996] McDonald, D. D. (1996). *Internal and External Evidence in the Identification and Semantic Categorization of Proper Names*, pages 21–39.

[Mendes et Antunes, 2009] Mendes, A. C. et Antunes, C. (2009). Pattern mining with natural language processing : An exploratory approach. *In International Conference of Machine Learning and Data Mining in Pattern Recognition (MDLM'09)*, pages 266–279.

[Messiant *et al.*, 2010] Messiant, C., Gábor, K. et Poibeau, T. (2010). Acquisition de connaissances lexicales à partir de corpus : la sous-catégorisation verbale en français. *Traitement Automatique des Langues (TAL)*, 51:65–96.

[Mikheev *et al.*, 1999] Mikheev, A., Moens, M. et Grover, C. (1999). Named entity recognition without gazetteers. *In Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*.

[Miller *et al.*, 2004] Miller, S., Guinness, J. et Zamanian, A. (2004). Name tagging with word clusters and discriminative training. *In Conference on Human Language Technology and North American chapter of the Association for Computational Linguistics (HLT/NAACL'04)*, pages 337–342, Boston, USA.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Education.

[Mooney et Bunescu, 2005] Mooney, R. J. et Bunescu, R. C. (2005). Mining knowledge from text using information extraction. *In ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD'05)*.

[Nadeau, 2007] Nadeau, D. (2007). *Semi-Supervised Named Entity Recognition : Learning to Recognize 100 Entity Types with Little Supervision*. Thèse de doctorat, University of Ottawa, Canada.

[Nadeau et Sekine, 2007] Nadeau, D. et Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30:3–26.

[Nouvel *et al.*, 2010a] Nouvel, D., Antoine, J.-Y., Friburger, N. et Maurel, D. (2010a). An analysis of the performances of the casen named entities recognition system in the ester2 evaluation campaign. *In International Language Resources and Evaluation (LREC'10)*.

[Nouvel *et al.*, 2011a] Nouvel, D., Antoine, J.-Y., Friburger, N. et Soulet, A. (2011a). Recognizing named entities using automatically extracted transduction rules. *In Language & Technology Conference (LTC'11)*.

[Nouvel *et al.*, 2012] Nouvel, D., Antoine, J.-Y., Friburger, N. et Soulet, A. (2012). Coupling knowledge-based and data-driven systems for named entity recognition. *In Innovative hybrid approaches to the processing of textual data (HYBRID'12, EACL Workshop)*.

[Nouvel *et al.*, 2011b] Nouvel, D., Antoine, J.-Y., Maurel, D. et Friburger, N. (2011b). Reconnaissance d'entités nommées, nouvelles frontières et nouvelles approches. Journée de l'ATALA - Reconnaissance d'Entités Nommées, Nouvelles Frontières et Nouvelles Approches.

[Nouvel et Soulet, 2011] Nouvel, D. et Soulet, A. (2011). Annotation d'entités nommées par extraction de règles de transduction. *In Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC'11)*.

[Nouvel *et al.*, 2010b] Nouvel, D., Soulet, A., Antoine, J.-Y., Friburger, N. et Maurel, D. (2010b). Reconnaissance d'entités nommées : enrichissement d'un système à base de connaissances à partir de techniques de fouille de textes. *In Traitement Automatique des Langues Naturelles (TALN'10)*.

[Ogden et Richards, 1923] Ogden, C. K. et Richards, I. A. (1923). *The Meaning of meaning*.

[Parekh et Honavar, 2000] Parekh, R. et Honavar, V. (2000). *Grammar Inference, Automata Induction, and Language Acquisition*, chapitre 29, pages 727–764.

[Pedregosa *et al.*, 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. et Édouard Duchesnay (2011). Scikit-learn : Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pei *et al.*, 2004] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U. et Hsu, M.-C. (2004). Mining sequential patterns by pattern-growth : The prefixspan approach. *In IEEE Transactions on Knowledge and Data Engineering*, volume 16, pages 1424–1440.

[Piaget et Chomsky, 1975] Piaget, J. et Chomsky, N. (1975). Seuil.

[Plantevit *et al.*, 2009] Plantevit, M., Charnois, T., Klema, J., Rigotti, C. et Cremilleux, B. (2009). Combining sequence and itemset mining to discover named entities in biomedical texts : a new type of pattern. *International Journal of Data Mining, Modelling and Management (IJDMM)*, 1:119–148.

[Poibeau, 2006] Poibeau, T. (2006). Dealing with metonymic readings of named entities. *In Annual Conference of the Cognitive Science Society (COGSCI'06)*.

[Qian *et al.*, 2010] Qian, X., Zhang, Q., Huang, X. et Wu, L. (2010). 2d trie for fast parsing. *In International Conference on Computational Linguistics (COLING'10)*, pages 904–912, Beijing, China.

[Raymond et Fayolle, 2010] Raymond, C. et Fayolle, J. (2010). Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement. *In Traitement Automatique des Langues Naturelles (TALN'10).*

[Riloff et Jones, 1999] Riloff, E. et Jones, R. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. *In National Conference on Artificial Intelligence (AAAI'99)*, pages 474–479.

[Roche, 2010] Roche, M. (2010). Filtrage des entités nommées par des méthodes de fouille de textes. *In Terminologie & Ontologie, Théories et applications (TOTH'10).*

[Rosset *et al.*, 2012] Rosset, S., Grouin, C., Fort, K., Galibert, O., Kahn, J. et Zweigenbaum, P. (2012). Structured named entities in two distinct press corpora : Contemporary broadcast news and old newspaper. *In Conference of the Association for Computational Linguistics (ACL'11) - Sixth Linguistic Annotation Workshop (LAW-VI)*, Jeju, South Korea.

[Russell, 1905] Russell, B. (1905). *On Denoting.*

[Savary *et al.*, 2010] Savary, A., Waszczuk, J. et Przepiórkowski, A. (2010). Towards the annotation of named entities in the national corpus of polish. *In International Language Resources and Evaluation (LREC'10).*

[Schmid, 1994] Schmid, H. (1994). Probabilistic pos tagging using decision trees. *In New Methods in Language Processing (NEMLP'94).*

[Srikant et Agrawal, 1996] Srikant, R. et Agrawal, R. (1996). Mining sequential patterns : Generalizations and performance improvements. *In International Conference on Extending Database Technology (EDBT'96)*, pages 3–17.

[Stephens, 1993] Stephens, C. S. (1993). The analysis and acquisition of proper names for the understanding of free text. *Computers and the Humanities*, 26:441–456.

[Stern et Sagot, 2010] Stern, R. et Sagot, B. (2010). Détection et résolution d'entités nommées dans des dépêches d'agence. *In Traitement Automatique du Langage Naturel (TALN'10).*

[Sun et Grishman, 2010] Sun, A. et Grishman, R. (2010). Semi-supervised semantic pattern discovery with guidance from unsupervised pattern clusters. *In International Conference on Computational Linguistics (COLING'10)*, pages 1194–1202, Beijing, China.

[Tan *et al.*, 2002] Tan, P.-N., Kumar, V. et Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. *In ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD'02)*, pages 32–41.

[Tran et Maurel, 2006] Tran, M. et Maurel, D. (2006). Prolexbase - un dictionnaire relationnel multilingue de noms propres. *Traitement Automatique des Langues (TAL)*, 47-3:115–139.

[van Rijsbergen, 1979] van Rijsbergen, K. (1979). *Information Retrieval.* Butterworth.

[Wang et Han, 1997] Wang, J. et Han, J. (1997). Bide : Efficient mining of frequent closed sequences. *Data Mining and Knowledge Discovery (DMKD)*, 1:259–289.

[Wang et Han, 2004] Wang, J. et Han, J. (2004). Bide : Efficient mining of frequent closed sequences. *In International Conference on Data Engineering (ICDE'04)*, pages 79–.

[Zhu *et al.*, 2007] Zhu, F., Yan, X., Han, J. et Yu, P. S. (2007). Efficient discovery of frequent approximate sequential patterns. *In IEEE International Conference on Data Mining (ICDM'07)*, pages 751–756.

[Zidouni *et al.*, 2009] Zidouni, A., Glotin, H. et Quafafou, M. (2009). Recherche d'entités nommées dans les journaux radiophoniques par contextes hiérarchique et syntaxique. *In Conférence en Recherche d'Informations et Applications (CORIA'09)*, pages 421–432.

[Zidouni *et al.*, 2010] Zidouni, A., Rosset, S. et Glotin, H. (2010). Efficient combined approach for named entity recognition in spoken language. *In Conference of the International Speech Communication Association (INTERSPEECH'10)*.

# Résumé :

Ces dernières décennies, le développement considérable des technologies de l'information et de la communication a modifié en profondeur la manière dont nous avons accès aux connaissances. Face à l'afflux de données et à leur diversité, il est nécessaire de mettre au point des technologies performantes et robustes pour y rechercher des informations. Les entités nommées (personnes, lieux, organisations, dates, expressions numériques, marques, fonctions, etc.) sont sollicitées afin de catégoriser, indexer ou, plus généralement, manipuler des contenus. Notre travail porte sur leur reconnaissance et leur annotation au sein de transcriptions d'émissions radiodiffusées ou télévisuelles, dans le cadre des campagnes d'évaluation Ester2 et Etape. En première partie, nous abordons la problématique de la reconnaissance automatique des entités nommées. Nous y décrivons les analyses généralement conduites pour traiter le langage naturel, discutons diverses considérations à propos des entités nommées (rétrospective des notions couvertes, typologies, évaluation et annotation) et faisons un état de l'art des approches automatiques pour les reconnaître. A travers la caractérisation de leur nature linguistique et l'interprétation de l'annotation comme structuration locale, nous proposons une approche par instructions, fondée sur les marqueurs (balises) d'annotation, dont l'originalité consiste à considérer ces éléments isolément (début ou fin d'une annotation). En seconde partie, nous faisons état des travaux en fouille de données dont nous nous inspirons et présentons un cadre formel pour explorer les données. Les énoncés sont représentés comme séquences d'items enrichies (morpho-syntaxe, lexiques), tout en préservant les ambigüités à ce stade. Nous proposons une formulation alternative par segments, qui permet de limiter la combinatoire lors de l'exploration. Les motifs corrélés à un ou plusieurs marqueurs d'annotation sont extraits comme règles d'annotation. Celles-ci peuvent alors être utilisées par des modèles afin d'annoter des textes. La dernière partie décrit le cadre expérimental, quelques spécificités de l'implémentation du système (mXS) et les résultats obtenus. Nous montrons l'intérêt d'extraire largement les règles d'annotation, même celles qui présentent une moindre confiance. Nous expérimentons les motifs de segments, qui donnent de bonnes performances lorsqu'il s'agit de structurer les données en profondeur. Plus généralement, nous fournissons des résultats chiffrés relatifs aux performances du système à divers point de vue et dans diverses configurations. Ils montrent que l'approche que nous proposons est compétitive et qu'elle ouvre des perspectives dans le cadre de l'observation des langues naturelles et de l'annotation automatique à l'aide de techniques de fouille de données.



# Mots clés :

Traitement automatique des langues, fouille de données, entités nommées, règles d'annotation.



# Abstract :

Those latest decades, the development of information and communication technologies has substantially modified the way we access knowledge. Facing the volume and the diversity of data streams, working out robust and efficient technologies to retrieve information becomes a necessity. In this context, Named Entities (persons, locations, organizations, numerical expressions, brands, functions, etc.) may be required in order to categorize, index or, more generaly, manipulate contents. Our work focuses on their recognition and annotation inside radio and TV broadcasts transcripts, in the context of Ester2 and Etape evaluation campaigns. In the first part, we introduce our problematic, the automatic recognition of named entities. We describe the commonly conducted analysis to process natural language, question the linguistic properties of named entities (related notions, typologies, evaluation and annotation) and describe state-of-the-art approaches. From their linguistic nature and by interpreting annotation as a local structuring, we propose an instruction-driven approach, based on annotation markers (tags), which originality consists in considering those elements in isolation. In the second part, we present the formalism used to explore data and introduce our formal framework. Sentences are represented as sequences of enriched items (morpho-syntax, lexicon) that preserve ambiguity. We also propose an alternative representation by segments that allows to limit combinatorial search. Patterns correlated to annotation markers are extracted as annotation rules. Those may be used by models so as to actually annotate texts. The last part presents the experimental framework, the implemented system (mXS) and the obtained results. We show the interrest of widely extracting annotation rules, even those of low confidence. We experiment segment patterns, that give interresting performances for deeply structured data. More generaly, we give results relative to performances of the system from diverse points of view and in diverse configurations. They show that the proposed approach is competitive and that it opens up perspectives for natural language observation and automatic annotation using data mining.



# Keywords :

Natural Language Processing, Named Entities, Data Mining, Annotation Rules.