Production d'annotations morpho-syntaxiques et d'expressions logiques et exploitation de données statistiques sur la suite FraCaS

Morgane Pailler

Août 2020

3ème année de Licence Mathématiques Informatique Appliquées aux Sciences Humaines et Sociales, Parcours Sciences Cognitives

Stage au sein de l'équipe Sémagramme - Loria & Inria NGE

Table des matières

I Sujet 4

II Le projet FraCaS et sa suite de tests 4

1 Définition de FraCaS 5
1.1 Genèse du projet 5
1.2 Nature de la suite de test 5

2 Adaptation de la suite en plusieurs langues : MultiFraCaS 5
2.1 FraCaS-FR 6

3 Observer comment des locuteurs natifs comprennent les inférences 8
3.1 Expérience menée avec la version française 8

III Organisation 9

IV Tâches effectuées 10

4 Participer au crowdsourcing 10

5 Proposer une expression sous forme logique des problèmes 10

6 Procéder à l'étiquetage morphosyntaxique et l'analyse des dépendances de la ressource 13

7 Répondre à une étude similaire et en étudier le rapport 16

8 Reprendre les scripts d'analyse statistiques de ma prédécesseur et les approfondir 17

9 Proposer de nouveaux problèmes d'inférence 26

V Autres évènements auxquels j'ai assisté pendant le stage 28

10 Conférence JEP-TALN-RECITAL 2020 28

11 Réunion avec l'équipe Sémagramme 28

VI Outils informatiques et conceptuels utilisés 29

12 Grew [1] **29**

13 Universal Dependencies et UD Pipe **30**
13.1 UDPipe[2] 30
13.2 Format CoNLL-U[3] 30

14 Arborator-Grew [4] **31**

15 Etherpadlite[5] **32**

16 Discord[6] **32**

17 Overleaf[7] **32**

18 GitLab [8] [9] **32**

19 Jupyter Notebook[10] **33**

VII Impressions sur le stage **33**

VIII Annexe **36**

20 Types d'inférences **36**

21 Propositions d'expressions logiques pour les inférences **38**

Introduction

J'effectue ce stage au sein de l'équipe Sémagramme dans le département de Traitement Automatique des Langues et Connaissances du Loria et de l'Inria Nancy-Grand Est. Mes tuteurs sont Maxime Amblard, maitre de conférences à l'Université de Lorraine et Bruno Guillaume, chargé de recherche à l'INRIA.

Première partie

Sujet

La suite de test FraCaS rassemble des problèmes d'inférence textuelle. L'objectif du stage est d'ajouter des annotations de différentes natures aux énoncés de la ressource. Il s'agit d'appliquer des outils existants pour produire des annotations morpho-syntaxiques et sémantiques (notamment à partir de certains outils développés par l'équipe Sémagramme).

Tâches confiées :

— Améliorer la traduction de la suite de tests en français,
— Prendre en main les différents outils de l'équipe Sémagramme : Grew, ACGtk
— Produire des annotations pour la ressource FraCaS : annotation en parties du discours, syntaxe de dépendances et structures sémantiques de la suite de test
— Tester la capacité des locuteurs à répondre correctement aux questions qu'elle pose

Compétences à acquérir ou à développer :

— Maîtrise d'outils informatiques
— Compréhension des phénomènes linguistiques modélisés
— Valider les annotations produites
— Découverte de la recherche, de ses méthodes et ses pratiques

Deuxième partie

Le projet FraCaS et sa suite de tests

Le principe global de FraCaS est de permettre de tester comment des problèmes sont interprétés par un système ou plus tard par des humains. Comme nous le verrons la suite a été conçue en suivant des règles logiques. Cependant son exploitation a rapidement mis en évidence des désaccords dans l'interprétation des problèmes entre les lecteurs. Ces ambiguïtés nous rappellent que l'usage courant de la langue n'est pas celui du logicien et FraCaS permet de mettre en évidence ces différences.

1 Définition de FraCaS

1.1 Genèse du projet

FraCas est une abréviation pour "Framework for Computational Semantics". Un Framework est un "cadre de travail", une architecture pour concevoir une famille d'applications. Il s'agit d'un projet européen de recherche mené entre 1994 et 1996 élaboré en collaboration avec le CWI d'Amstredam, le SIR de Cambridge, l'Université d'Edimbourg (Centre for Cognitive Science), l'Université de Saarbrücken (Computational Linguistics) et l'Université de Stuttgart (IMS). FraCas est une suite de test utilisée pour mesurer les compétences sémantiques d'un système de TAL (Traitement Automatique des Langues). Elle est initialement présentée au chaptire 3 de l'article *Using the Framework*[11].

1.2 Nature de la suite de test

La suite de test FraCaS prend la forme de 346 problèmes d'inférences en langage naturel (en anglais dans la version originale).

D'après le Larousse[12] par inférence on entend, "Opération par laquelle on passe d'une assertion considérée comme vraie à une autre assertion au moyen d'un système de règles qui rend cette deuxième assertion également vraie".

Exemples d'inférences : Texte 1 : *L'argent ne fait pas le bonheur.*
Hypothèse : *Un homme riche peut être malheureux.*

Texte 2 : *Tous les enfants doivent aller à l'école et Jean est un enfant*
Question : *Jean doit-il aller à l'école ?*

Pour FraCaS, les problèmes ont été conçus de manière artificielle pour illustrer des phénomènes sémantiques.

A l'origine il teste la capacité de systèmes de TAL à faire de la reconnaissance de relations entre des fragments de texte (en anglais "textual entailment recognition"(TER)). C'est à dire en quelques sortes ici à "comprendre" les inférences proposées.

FraCaS propose des exemples permettant de tester la capacité d'un système à reconnaître si d'une entrée T (un ensemble d'1 à 5 prémisses) un humain aurait en général pu inférer une certaine $Hypothese$ (présentée dans la version originale sous la forme de questions).

Les problèmes sont répartis en 9 catégories détaillées en annexe (Inférences basées sur des quantificateurs, des ellipses, des anaphores...)

2 Adaptation de la suite en plusieurs langues : MultiFraCaS

FraCaS a été utilisée pour des systèmes de compréhension, de logique naturelle et d'assistants de preuve. Comme l'indique *Testing the FraCaS test suite* [13] la ressource a également été traduite dans plusieurs langues : MultiFraCas est un projet qui a permis la création de tests équivalents entre autre en Farsi, Allemand, Grec et Mandarin.

2.1 FraCaS-FR

Et nous allons plus particulièrement nous concentrer sur la version sur laquelle porte ce stage : la version française. Celle ci a été développée au sein de l'équipe Sémagramme par Maxime Amblard, Clément Beysson, Philippe de Groote, Bruno Guillaume et Sylvain Pogodalla. Les objectifs étaient notamment de pouvoir bénéficier des comparaisons possibles entre les différentes versions multilingues, tester certains phénomènes logiques et sémantiques et collecter des données (pour l'instant rares) sur le français quant aux inférences (d'après *A French Version of the FraCaS Test Suite* [14]).

Problèmes rencontrés dans la traduction de FraCaS en français

La première version de FraCaS-FR avait pour but de s'adapter le plus possible au lexique et à la syntaxe originale mais cela a mené à des formulations parfois étranges. D'ailleurs il est intéressant de noter que comme pour les différentes versions de MultiFraCaS les traductions ne sont pas littérales car cela ne permettait pas forcément de reconnaître tous les phénomènes sémantiques de la version originale en anglais. De plus on ne retrouve de toute façon pas forcément d'équivalent à tous ces phénomènes dans les autres langues. Mais, en tout cas pour la version française, toutes les 346 inférences (même celles ne contenant pas à proprement parler de problème) ont été traduites. Voici quelques problèmes rencontrés dans la traduction

Première source de problèmes : Plusieurs fois le même problème dans la suite FraCaS originale

Dans tout le document P signifiera Prémisse, Q : Question et H : Hypothèse

Exemple : Problèmes 87 et 88
P Tous les représentants et les clients étaient à la réunion.
H Tous les représentants étaient à la réunion.

La seule différence entre ces problèmes est la réponse attendue "Oui" pour une lecture et "Pas assez d'informations" pour l'autre.

Deuxième source de problèmes : Traduction identique de problèmes différents en anglais

Exemple : Problèmes 108 et 110 (décrits dans *A French Version to the FraCaS Suite*[14])
Problème 108 : Version originale

P Just one accountant attended the meeting.
Q Did **any** accountant attend the meeting ?
H Some accountant attended the meeting.

Problème 110 : Version originale
P Just one accountant attended the meeting.
Q Did **some** accountant attend the meeting ?
H Some accountant attended the meeting.

Traduction des deux problèmes :
P Un seul comptable a assisté à la réunion.
Q Un comptable a-t-il assisté à la réunion ?
H Un comptable a assisté à la réunion.

Troisième source de problèmes : Cas où des variations dans la position de l'adverbe dans le problème original donnent lieu à des interprétations différentes en anglais mais pas en français[14]

Exemple : Problèmes 255 et 256
Problèmes 255 : Version originale
P1 Since 1992 ITEL has made a loss.
P2 It is now 1996.
Q Did ITEL make a loss in 1993 ?
H ITEL made a loss in 1993.
En Français
P1 Depuis 1992 ITEL a subi des pertes.
P2 Nous sommes maintenant en 1996.
Q ITEL a-t-elle subi des pertes en 1993 ?
H ITEL a subi des pertes en 1993.
Problèmes 256 : Version originale
P1 ITEL has made a loss since 1992.
— ...
En Français
P1 ITEL a subi des pertes depuis 1992.
— ...

3 Observer comment des locuteurs natifs comprennent les inférences

Une démarche pour vérifier les inférences par une production participative (crowdsourcing) en ligne a été mise en place pour plusieurs versions de FraCaS.

D'abord dans une étude pilote un échantillon de 15 problèmes de FraCas représentatifs de chaque genre de problèmes (voir Types d'inférences) ont été choisis.

Les testeurs ont pour instruction de "répondre à la question en se basant seulement sur les informations données dans les phrases. Par exemple si les informations sont contenues dans les phrases on pourra répondre à la question par 'oui' ou 'non' sinon la réponse sera 'Je ne sais pas'. En répondant à ces questions vous ne devez vous baser sur aucune autre connaissance qui ne soit pas déclarée dans les phrases."

Trois études sont présentées dans *Testing the FraCaS test suite* [13] :

— Etude 1 : un petit groupe de 14 étudiants anglophones (natifs ou suédois avec un excellent niveau en anglais) familiers avec la résolution de ce genre de problèmes.

— Etude 2 : Un groupe ouvert de 46 employés de l'université de Göteborg anglophones (natifs ou suédois avec un excellent niveau en anglais) pas habitués à résoudre ce genre de problèmes.

— Etude 3a et 3b : (MultiFraCas) Un groupe de 57 amis parlant grec et un groupe de 29 amis parlant slovène choisis au hasard sur les réseaux sociaux pas habitués à résoudre ce genre de problèmes.

On peut constater des désaccords (et donc des ambiguïtés) dans les réponses ne correspondant pas aux attentes par exemple pour le problème 201 dont la réponse est supposée être "Don't know" mais dont le nombre réponses justes pour cette question ne dépasse jamais la moitié des interrogés (maximum avec le groupe slovène avec seulement environ 50 % de "Don't know").

Problème 201 :
P1 John is a former successful university student
Q Is John a university student ?
H John is a university student.

3.1 Expérience menée avec la version française

Une démarche similaire a été menée avec la traduction française. Une interface web accessible sur inscription a été mise en place et les problèmes sont présentés de telle façon :

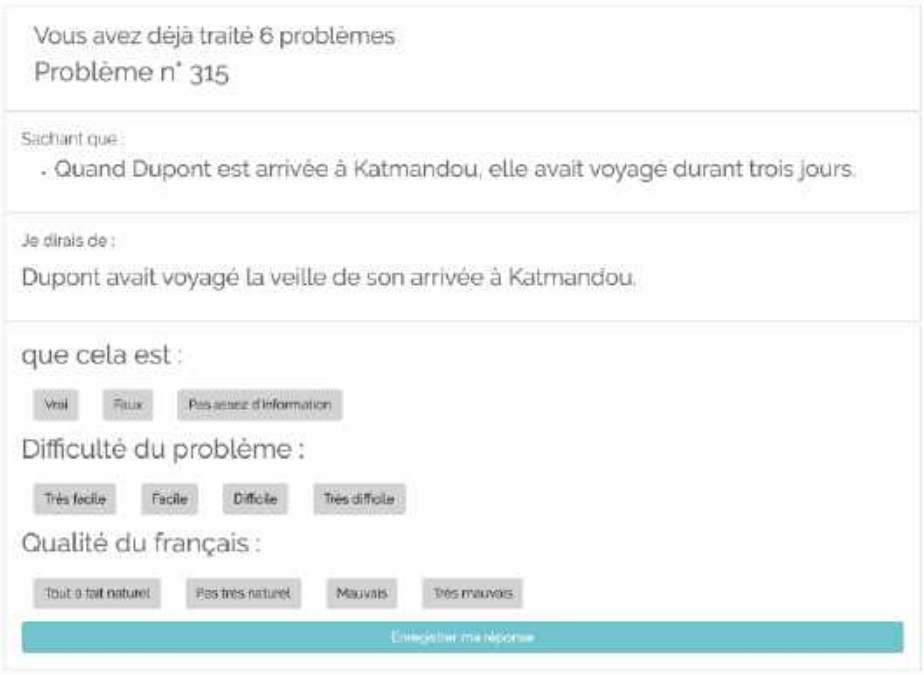

FIGURE 1 – Capture d'écran tirée de *A French Version of the FraCaS Test Suite*. "Sachant que *T* je dirais que *Hypothese* (sous une forme déclarative). "

342 problèmes y sont affichés (4 des 346 inférences originales n'étant pas vraiment des problèmes).

Après quoi les participants doivent également préciser leur ressenti sur la difficulté (sur une échelle de 0 à 3 allant de Très facile à Très difficile) et la qualité du français, de la traduction (de 0 à 3 allant de Tout à fait naturel à Très mauvais). Ce site a été diffusé notamment parmi les stagiaires de l'équipe Sémagramme et leur entourage.

Seules les personnes ayant répondu à tous les problèmes (7 au moment de la publication de *A French Version of the FraCaS Test Suite* [14]) pouvaient voir leurs annotations prises en compte dans les statistiques de comparaison avec les réponses attendues. Les réponses des personnes ayant répondu à au moins 10% des problèmes avaient leurs réponses prises en compte pour la qualité et la complexité des énoncés (12 au moment de la publication).

Troisième partie

Organisation

En raison des mesures sanitaires liées à l'épidémie de la Covid-19 j'ai donc effectué mon stage en télétravail.

Dans ces conditions, chaque semaine était planifiée une réunion permettant de faire un retour sur les tâches effectuées et de réfléchir à celles à suivre. Celle ci se faisait donc en visioconférence avec l'application Discord. Cela nous a permis d'effectuer des partages d'écrans pour présenter le travail effectué dans la semaine. Ces réunions duraient en moyenne 1h30.

Les échanges se faisaient d'autre part par mail régulièrement entre les réunions en cas de questions. Je devais par ce biais également présenter en amont les points à aborder dans la prochaine réunion afin d'avoir en tête le déroulé et ne pas perdre de temps à chercher de quoi parler. D'une certaine façon c'était déjà un bon exercice de communication.

Nous avions également un éditeur de texte collaboratif hébergé par l'Université (avec pour client Etherpadlite) qui nous a servi à avoir en commun les dates (début et fin du stage, réunions, rendus) tâches et ressources (outils, bibliographie) clés du stage.

Enfin on m'a encouragé à utiliser Overleaf (autre éditeur de texte permettant de travailler à plusieurs sur un document mais ayant des fonctionnalités intéressantes) comme carnet de bord, support pour mes notes et donc base pour mon rapport. Je l'ai également employé pour écrire les problèmes sous forme logique pour pouvoir travailler sur celles-ci à plusieurs durant les réunions.

Pour toutes ces tâches, j'ai utilisé mon ordinateur personnel.

Je suis restée dans ma région d'origine et n'ai donc pas rencontré mes tuteurs durant le stage.

En terme d'organisation de mon travail, j'avais des tâches hebdomadaires récurrentes telles que la proposition d'expression logique pour les inférences et en fin de réunion nous prenions toujours le temps de me fixer des objectifs pour la semaine. En début de stage nous avions également défini des tâches annexes que je pouvais réaliser si j'en avais le temps (ex : les proposition de nouvelles inférences).

Quatrième partie
Tâches effectuées

4 Participer au crowdsourcing

Une de mes premières tâches a été, comme les sujets des expériences décrites dans la bibliographie, de participer au questionnaire portant sur les inférences. Une erreur de me part a été de d'abord lire la bibliographie et regarder le fichier XML de la ressource ce qui (bien que je n'ai pas retenu toutes les réponses bien entendu) a pu biaiser mes réponses.

5 Proposer une expression sous forme logique des problèmes

La logique derrière les inférences de la suite de tests est une dimension importante de ces problèmes. C'est pourquoi il a semblé intéressant de produire à partir de ces problèmes une retranscription en langage/logique. En effet il serait intéressant d'avoir un corpus en langage logique avec sa transcription.

On m'a confié donc la tâche de chaque semaine reformuler en expressions logiques environ 5 problèmes issus de FraCaS (en prenant d'abord tous les problèmes se terminant par 0).

Exemple de reformulation en logique : **Problème 20**

P1 Tout Européen a le droit de vivre en Europe.

P2 Tout Européen est une personne.

P3 Toute personne qui a le droit de vivre en Europe peut circuler librement en Europe.

Hyp Tout Européen peut circuler librement en Europe.

Rep Vrai

Formulation logique :

— $europeen(x)$: x est européen

— $personne(x)$: x est une personne

— $vivre\_europe(x)$: x vit en Europe

— $circuler\_librement\_europe(e)$: x circule librement en Europe

P1 $\forall x \exists e.europeen(x) \rightarrow vivre\_europe(e) \wedge \Diamond Agent(e, x)$

P2 $\forall x.europeen(x) \rightarrow personne(x)$

P3 $\forall x \exists e \exists e'.personne(x) \wedge vivre\_europe(e) \wedge \Diamond Agent(e, x)) \rightarrow (circuler\_librement\_europe(e') \wedge \Diamond Agent(e', x)$

Hyp $\forall x \exists e.europeen(x) \rightarrow circuler\_librement\_europe(e) \wedge \Diamond Agent(e, x)$

A chaque réunion hebdomadaire, nous prenions le temps de revenir sur les expressions logiques produites durant la semaine. Ceci à la fois pour essayer de les améliorer mais aussi pour discuter ensemble d'une norme que je devrais appliquer pour les problèmes afin d'avoir une expression cohérente et homogène.

Dans un premier temps mon objectif pour cette retranscription logique était de traiter 10% des problèmes, les problèmes dont l'identifiant se terminait par zero, soit 34 problèmes. J'ai terminé cette tâche fin juin.

Un aspect intéressant a été de pouvoir associer la rédaction des problèmes en logique avec les annotations que les personnes ayant participé à l'expérience avaient pu donner. C'est-à-dire parfois donner plusieurs expressions logiques d'un même énoncé pour en représenter les différentes interprétations possibles.

Exemple : extrait du Problème 170

P Jean a trouvé Marie avant Guillaume.

Formulation logique :

- JEAN
- GUILLAUME
- MARIE
- $trouver(e)$
- $time(e, t)$
- $avant(t, t')$: t précède t'

Pour une lecture : Jean a trouvé Marie avant que Guillaume trouve Marie.

P $\exists e \exists e' \exists t \exists t'.trouver(e) \wedge Agent(e, \text{JEAN}) \wedge Patient(e, \text{MARIE}) \wedge trouver(e') \wedge Agent(e', \text{GUILLAUME}) \wedge Patient(e', \text{MARIE}) \wedge time(e, t) \wedge time(e', t') \wedge avant(t, t')$

Pour une seconde lecture : Jean a trouvé Marie avant que Jean trouve Guillaume.

P $\exists e \exists e' \exists t \exists t'.trouver(e) \wedge Agent(e, \text{JEAN}) \wedge Patient(e, \text{MARIE}) \wedge trouver(e') \wedge Agent(e', \text{JEAN}) \wedge Patient(e', \text{GUILLAUME}) \wedge time(e, t) \wedge time(e', t') \wedge avant(t, t')$

Mais également réfléchir au niveau de finesse requis pour décrire le problème sans le rendre absurdement complexe mais en même temps permettre d'en saisir les ambiguïtés. Par exemple pour une phrase simple telle que "Jean écrit un livre" on va avoir une analyse assez fine du type : "$\exists x \exists e.livre(x) \wedge ecrire(e) \wedge Agent(e, \text{JEAN}) \wedge Patient(e, x)$".

Tandis que pour une phrase plus complexe telle que dans le problème 320 "Lorsque Durand a obtenu son emploi à la CIA, il savait qu'il ne serait jamais autorisé à écrire ses mémoires." on va se contenter pour retranscrire "écrire se mémoires" d'utiliser des formules du type $ecrire\_ses\_memoires(e)$ pour simplifier

Il fallait aussi porter une attention particulière à garder une certaine uniformité sur l'ensemble des expressions. Par exemple garder la même façon de représenter une caractéristique (par exemple "être une femme" $\rightarrow femme(x)$). Le tout pour garder une certaine cohérence.

Difficultés rencontrées

L'une des contraintes pour traiter plus de problèmes était la difficulté d'un consensus pour l'expression logique. En effet si chaque semaine nous prenions le temps de traiter 5 problèmes cela nous prenait souvent une quarantaine de minutes sur des réunions d'1h30. Plus encore, le mercredi 8 juillet nous avons fait une réunion avec d'autres personnes ayant travaillé sur le projet FraCaS-FR (Philippe de Groote et Sylvain Pogodalla). Cette réunion devait être un retour de la part de MM. de Groote et Pogodalla sur nos choix d'expression logique pour les 34 premiers problèmes retranscrits. Si cette séance a été très intéressante et enrichissantes au bout de 2h nous n'avions fait à peine travaillé que 3 problèmes et les façons d'exprimer les problèmes étaient (comme à notre attente) complètement remises en question.

Exemple de problème transformé durant la réunion : Problème 10
P La plupart des grands ténors sont italiens.
Hyp Il y a de grands ténors qui sont italiens.
Proposition d'expression logique par MM. Amblard, Guillaume et moi-même :
— L_T : La plupart (on crée un quantificateur)
— $grand\_tenor(x)$: x est un grand ténor
— $italien(x)$: x est italien
P : $L_T x, grand\_tenor(x) \rightarrow italien(x)$
Hyp : $\exists x, grand\_tenor(x) \wedge italien(x)$
Expression logique telle que révisée durant la réunion
— sémantique de $MOSTx(Ax)(Bx)$: $2x|A \wedge B| > |B|$
— P : $MOSTx(GrandTenorx)(Italienx)$
— Hyp : $\exists x, GrandTenorx \wedge Italienx$

J'avais alors commencé à retranscrire un autre lot de problèmes (choisis car présentant un fort désaccord dans les interprétations pour les participants à l'étude). Cependant j'ai après cette réunion arrêté car cela ne se révélait que peu utile. En effet ce que j'avais produit jusque là devait subir une totale remise en question.

Cette tâche m'a permis de mettre en pratique les connaissances en logique que j'ai pu acquérir pendant la licence mais aussi d'en apprendre de nouvelles.

6 Procéder à l'étiquetage morphosyntaxique et l'analyse des dépendances de la ressource

Utiliser différents outils et en comparer les résultats

Le première étape de l'étiquetage a été d'utiliser deux outils différents pour procéder à l'annotation de la ressource (Grew et UDPipe, voir dans la section Outils utilisés) et ensuite d'observer les différences dans leurs résultats. On pouvait constater par exemple un niveau de finesse différent dans l'annotation (par exemple un pronom "Il" décrit comme sujet chez GREW et comme sujet explétif chez UDPIPE).

Après observation d'un échantillon de phrases nous avons choisi de nous baser sur l'annotation de UDPipe tout en observant les différences avec Grew pour y repérer un premier lot d'erreurs. Notamment pour les questions de segmentation.

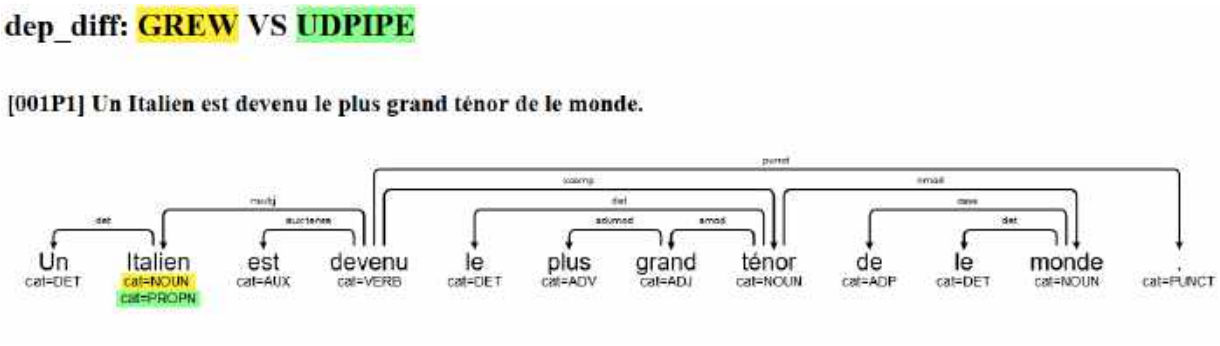

FIGURE 2 – Différence d'annotation des outils Grew et UDPipe

Faire des choix de segmentation

Pour un outil d'analyse syntaxique une première étape est la segmentation du corpus c'est-à-dire : Transformer un texte : "Je mange une pomme" En une liste d'unités lexicales (appelées token) : ["Je", "mange", "une", "pomme"] auxquels on pourra associer ensuite une étiquette morphosyntaxique : [("Je",PRON), ("mange",VERB), ("une",DET), ("pomme",NOUN)]

En français cela se complexifie notamment lorsqu'on trouve des contractions telles que à+les => aux. Les outils sont donc pensés pour prendre cela en compte et segmenter dans leur lecture le mot (aux) pour en avoir une analyse plus fine ([('à',ADP), ('les',DET)]).

Cependant les choses se compliquent pour 'des' et 'du'. On a en effet il faut faire un choix de segmentation entre des homonymes, d'une part **l'article indéfini pluriel des** et d'autre part **la contraction de la préposition de et de l'article les** et donc entre deux sens et usages différents (de même pour du article partitif du ou la contraction de et le). J'ai également vérifié ce même genre de confusion pour **"de"** pouvant être un article indéfini ("De grands ténors sont italiens").

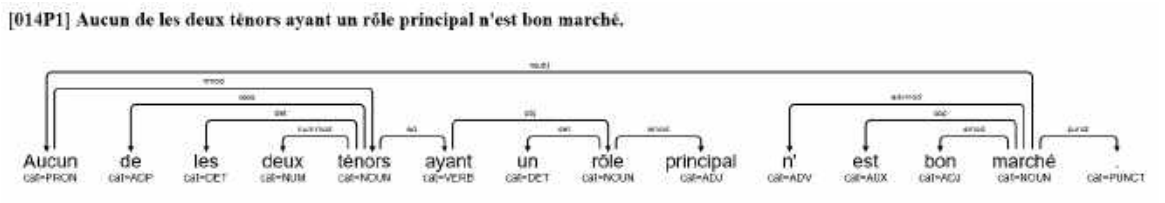

FIGURE 3 – Choix de segmentation entre 'de les' et 'des', exemple avec 'de les'

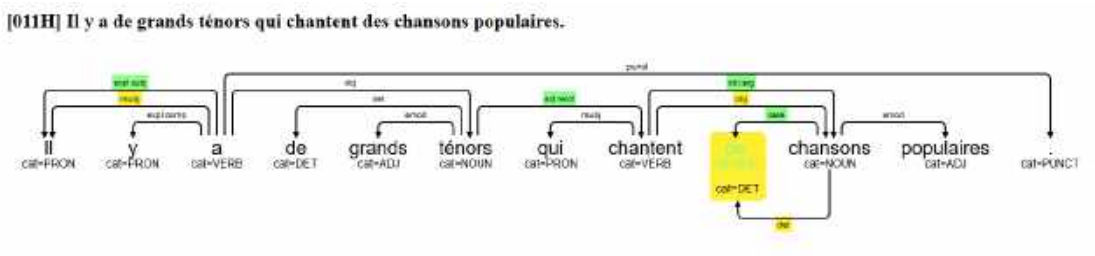

FIGURE 4 – Choix de segmentation entre 'de les' et 'des', exemple avec 'des'

Repérer des erreurs d'annotation récurrentes

Dans certains cas, notamment avec l'usage de noms propres ou de noms communs commençant par une majuscule comme les gentilés en français il peut y avoir des problèmes d'annotation (mot inconnu, non reconnu par le programme) qui peuvent également impacter l'étiquetage d'autres mots. Voici deux exemples d'annotations erronées récurrentes dans FraCaS : **Tout Européen**

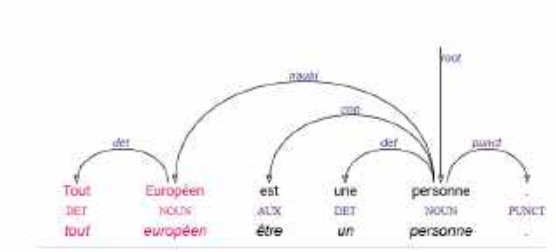

FIGURE 5 – Annotation correcte : Européen annoté comme un nom commun

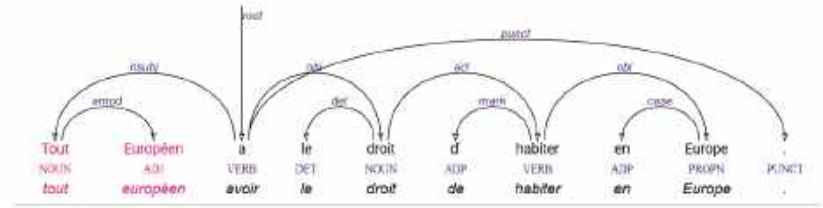

FIGURE 6 – Européen annoté comme étant un adjectif

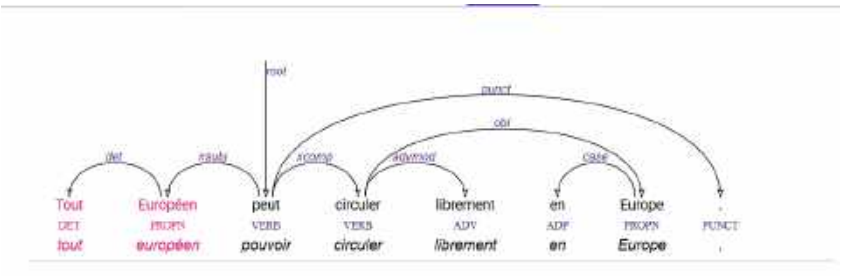

FIGURE 7 – Européen annoté comme étant un nom propre

ITEL

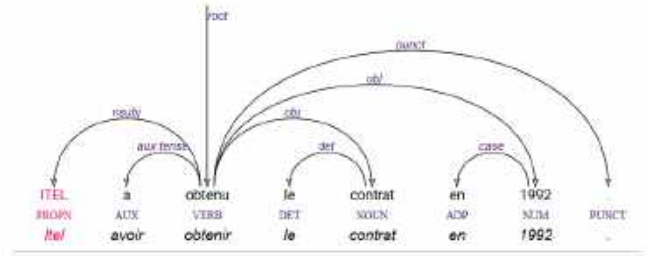

FIGURE 8 – Annotation correcte : ITEL annoté comme un nom propre

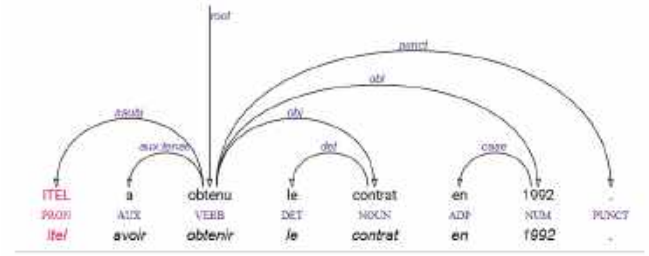

FIGURE 9 – Annotation incorrecte : ITEL annoté comme étant un pronom

7 Répondre à une étude similaire et en étudier le rapport

Durant le stage M. Amblard m'a transmis ainsi qu'aux autres étudiants de la Licence une étude[15] assez similaire au projet de test de la suite FraCaS. Il s'agit également d'une étude participative ouverte à tous (crowdsourcing) composée d'inférences présentant des ambiguïtés portant

plus spécifiquement sur les quantificateurs. Le projet a été mené par des étudiants en L3 Informatique à l'Université de Montpellier.

La démarche était similaire et il était intéressant de voir des façons différentes d'exploiter les données . Par exemple la recherche de groupes de tendances dans les réponses : "clustering") mais qui ne s'appliquait pas exactement à notre cas car notre nombre de réponses était plus limité.

En la lisant cela m'a fait pensé qu'il était pertinent (comme cela était fait dans l'étude) de travailler sur un petit lot de problèmes (ici une vingtaine contre plus de 300 dans FraCaS) pour étudier les résultats sur un échantillon d'invidus plus large (engagement plus facile pour 20 questions que pour 300).

Finalement cette ouverture m'a assez peu servi mais la tâche était intéressante. Puisque cette suite existait déjà, cela m'a encouragée à ne pas trop me concentrer sur les quantificateurs dans mes propositions d'inférences en français (redondant avec cette étude) .

8 Reprendre les scripts d'analyse statistiques de ma prédécesseur et les approfondir

L'étude participative a permis de récolter des données sur les réponses des annotateurs. Celles-ci étaient stockées dans une base de données (dont on m'a fait un export sous format `.CSV` pour pouvoir en exploiter les résultats plus simplement). La base de donnée est organisée comme suit : Pour chaque réponse à l'étude, on a :

- L'identifiant de la réponse
- L'identifiant du participant
- L'identifiant du problème
- La réponse sous format numérique correspondant à 0 : Oui, 1 : Non, 2 : Pas assez d'informations
- La qualité du problème estimée par le participant sous format numérique correspondant à 0 : Très mauvais, 1 : Mauvais, 2 : Pas très naturel, 3 : Tout à fait Naturel
- La complexité du problème estimée par le participant sous format numérique correspondant à 0 : Très facile, 1 : Facile, 1 : Difficile, 0 : Très difficile
- La date de la réponse

Étude de l'accord inter-annotateur

En cherchant des pistes de réflexions autour des données statistiques disponibles, j'ai relu l'étude et notamment des tableaux montrant des problèmes dans la version multilingue de la suite FraCaS avec des réponses très variables. Cela m'a donc donné l'idée à mon tour de mesurer l'accord inter-annotateur pour chaque problème. Donc pour un problème *key* à quel niveau les annotateurs étaient ils d'accord sur la réponse.

Dans l'étude *A French Version of the FraCaS Test Suite*[14], l'accord inter-annotateur global a déjà été mesuré. Par accord inter-annotateur on entend la proportion de cas où pour une même question les annotateurs observés deux à deux ont donné la même réponse. Pour cela les outils de la librairie NLTK sur Python ont été utilisés. C'est pourquoi M. Guillaume m'a conseillé de les utiliser comme indicateur de l'accord.

J'ai dans un premier temps voulu utiliser plus précisément le coefficient Kappa de Cohen comme cela a été fait dans l'étude. Mais ayant rencontré des difficultés à le mettre en place, je l'ai remplacé par la fonction accord moyen observé de NLTK (avg_Ao()). L'accord observé correspond à la proportion des cas où les participants sont d'accords.

Cette fonction a pour désavantage de ne pas prendre en compte la probabilité d'un accord aléatoire contrairement au Kappa. Par accord aléatoire on entend le fait que deux annotateurs même s'ils donnent leur réponse au hasard ont une certaine probabilité de donner la même réponse.

Si j'avais cependant réussi à utiliser le Kappa, il aurait été intéressant de prendre en compte qu'étant donné la façon dont est conçue la suite de test, on a un déséquilibre dans le nombre de Oui dans les réponses (ce qui doit influencer les annotateurs).

```
from nltk.metrics.agreement import AnnotationTask
agreement={}
for key in annotations :
    t= AnnotationTask(data=annotations[key])
    agreement[key] = t.avg_Ao()
```

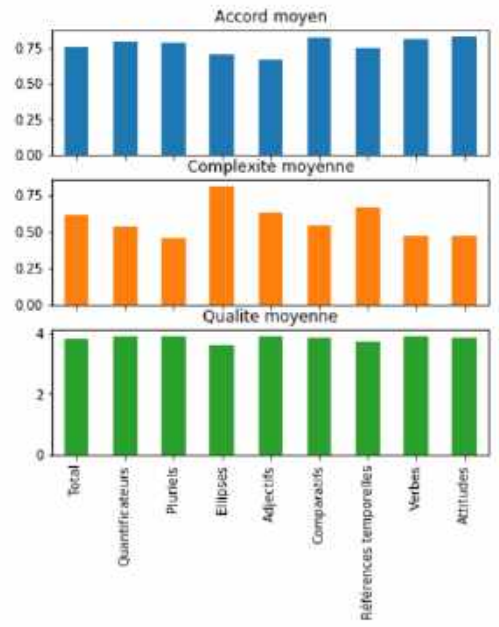

FIGURE 10 – Valeurs observées selon les catégories pour l'accord inter-annotateur (accord moyen donné par NLTK), la complexité et la qualité moyennes

Les différentes catégories en abscisse correspondent aux différents types d'inférences explicités au début de se rapport, autrement dit les différents phénomènes linguistiques réalisés dans les problèmes de FraCaS. Il est intéressant de remarquer les variations dans ces valeurs. Ces données ont permis d'étudier plusieurs hypothèses quant aux problèmes et annotations.

Hypothèses étudiées

Pour les graphiques suivants, le gradient de couleur illustrera la répartition des valeurs (plus le bleu est foncé plus on a d'annotations correspondant à ces valeurs) . En rouge sera représentée la régression linéaire de la complexité en fonction de la qualité. Le coefficient de Pearson représente la corrélation entre deux variables. Il varie entre -1 et 1 et la corrélation est considérée comme forte quand sa valeur absolue dépasse 0,5. La ligne sur les graphiques représentant la corrélation, matérialise le seuil pour lequel ces valeurs sont significatives.

Hypothèse 1 : Il y a corrélation entre la qualité de traduction d'un problème et sa complexité

Par qualité, on entend ici la qualité globale du français, de la traduction des énoncés. Donc ici on cherche à savoir si plus un problème est considéré comme complexe par les annotateurs, plus également il est considéré par eux comme mal traduit, mal formulé.

Cependant notons de prime abord que la qualité de la traduction de FraCaS-FR est plutôt bonne et assez homogène (moyenne élevée de 3,8/4 et une variance plutôt faible de 0,08 (seuls 8 problèmes ont une évaluation moyenne de leur traduction inférieure à 3)) ce qui n'offre pas vraiment la possibilité d'observer la complexité pour des traductions médiocres.

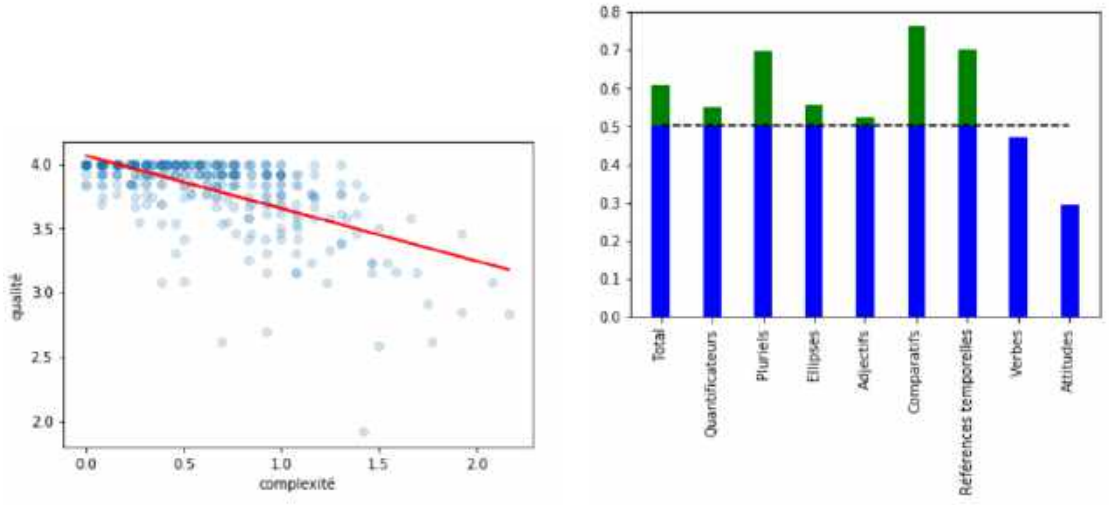

FIGURE 11 – A gauche une représentation graphique de **la complexité en fonction de la qualité** (Valeurs moyennes pour chaque problème). Sur le graphique de droite les valeurs absolues du coefficient de Pearson par catégories de problèmes(Valeur réelle moyenne : -0.609941)

On peut constater une certaine corrélation entre la qualité et la complexité moyennes d'un problèmes, sauf pour les inférences portant sur des Attitudes et des Verbes (voir Types d'inférences). De plus étant donné que le coefficient est négatif, on peut dire que lorsque la qualité est basse on voit la complexité augmenter (sans pouvoir préjuger bien sûr d'une causalité).

Dans un second temps il m'a cependant paru intéressant de voir si on pouvait retrouver cette corrélation pour une annotation donnée. C'est-à-dire savoir si, lorsque que quelqu'un évalue la

qualité comme mauvaise, en général, il considère la complexité comme élevée (pas seulement en terme de tendances des complexités et de la qualités moyennes).

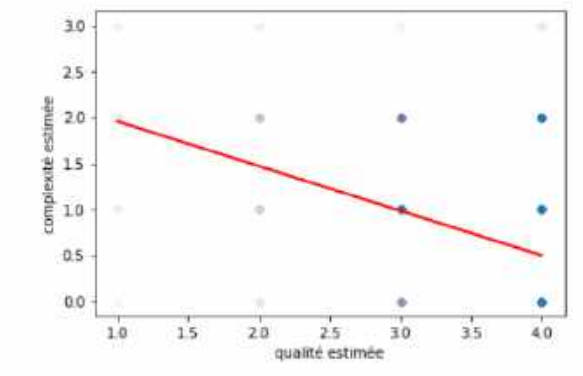

FIGURE 12 – Représentation graphique de **la complexité en fonction de la qualité** estimées pour chaque annotation. Valeur du coefficient de Pearson : -0.354179

On constate ici une corrélation faible entre la qualité et la complexité. Nous n'avons donc pas confirmation qu'il y ait corrélation entre la qualité et la complexité estimée d'un problème par un annotateur.

Ces divergences dans les résultats sont intéressantes car elles supposent que si un annotateur donné ne va pas forcément considérer un problème compliqué pour lui comme de mauvaise qualité, on pourra cependant repérer une tendance globale de relation entre ces deux variables.

Conclusion : Il est difficile dans ce cas de dire qu'on confirme l'hypothèse 1 "Il y a corrélation entre la qualité de traduction d'un problème et sa complexité".

Les hypothèses suivantes portent quand à elles sur une recherche et une évaluation des facteurs qui pourraient amener de possibles interprétations différentes de phrases, des ambiguïtés, des désaccords ou pas entre les annotateurs et donc une variation de l'accord inter-annotateur.

Hypothèse 2 : Il y a corrélation entre la qualité estimée et l'accord inter-annotateur

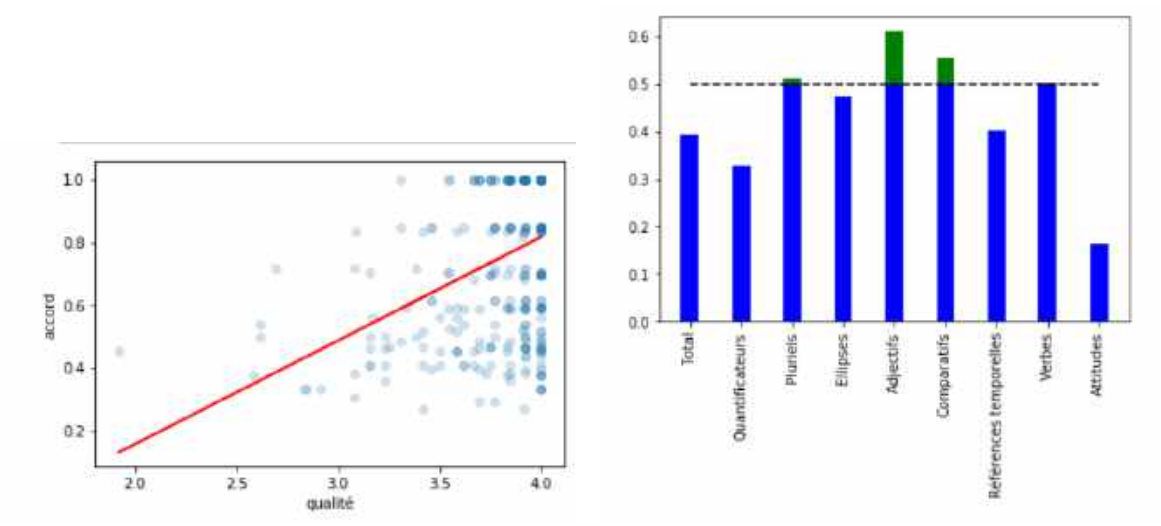

FIGURE 13 – A gauche une représentation graphique de **l'accord inter-annotateur en fonction de la qualité estimée moyenne**. A droite les **coefficient de corrélation selon les catégories** (valeur absolue du coefficient de Pearson) entre l'accord inter-annotateur et la qualité.

On observe une assez faible corrélation entre la complexité et l'accord inter-annotateur sur cet échantillon à part pour certaines catégories (Adjectifs en Comparatifs).

Conclusion : Les résultats ne nous permettent pas de confirmer un lien entre la qualité de traduction des problèmes et l'accord inter-annotateurs. A noter encore une fois la bonne qualité de la traduction et son manque de variation.

Hypothèse 3 : Il y a corrélation entre la complexité et l'accord inter-annotateur

En d'autres termes on cherche à savoir si des ambiguïtés dans les inférences rendent le problème complexe ou encore si la complexité d'un problème le rend ambigu. Il s'agit de vérifier une intuition : si un problème est complexe il semble qu'il y ait plus de chances pour que les annotateurs se "trompent" ou en tout cas divergent dans leurs interprétations de l'énoncé.

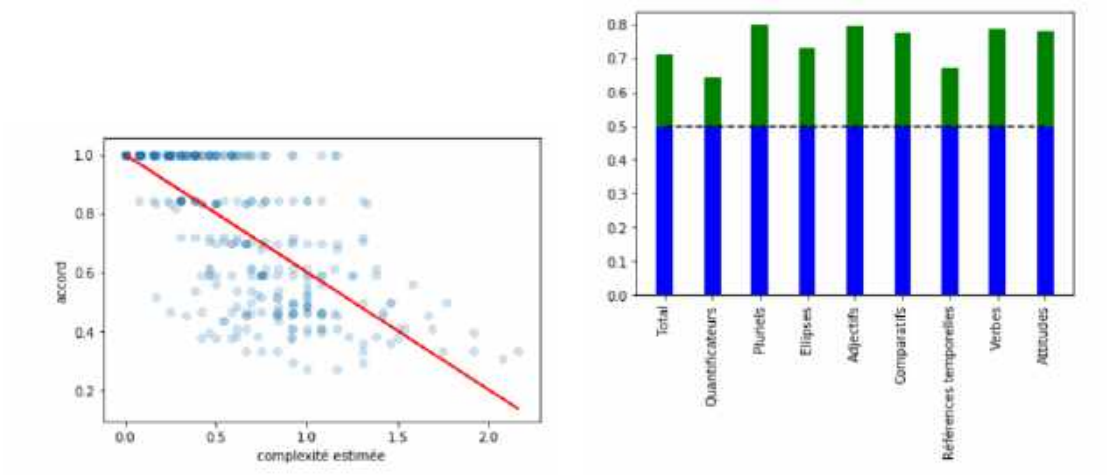

FIGURE 14 – A gauche une représentation graphique de **l'accord inter-annotateur en fonction de la complexité estimée moyenne**. A droite les **coefficient de corrélation selon les catégories** (valeur absolue du coefficient de Pearson) entre l'accord inter-annotateur et la complexité.

On constate une forte corrélation entre la complexité estimée et l'accord inter-annotateur. **Conclusion :** Les résultats corroborent l'hypothèse 3 "Plus un problème est considéré comme complexe, plus l'accord inter-annotateur baisse". Si on ne peut confirmer une causalité en plus d'une corrélation, on peut imaginer qu'une ambiguïté puisse être considérée comme une source de complexité par les annotateurs.

Hypothèses 4 et 5 : Il y a corrélation entre le nombre de prémisses et l'accord inter-annotateur et entre le nombre de prémisses et la complexité

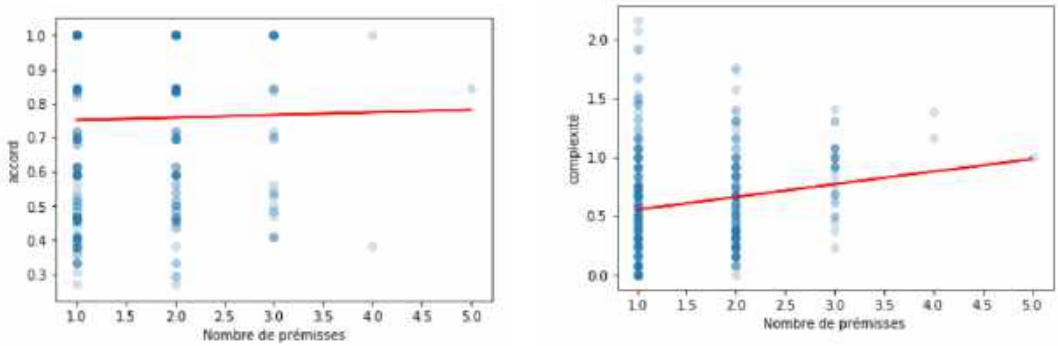

FIGURE 15 – A gauche une représentation graphique de **l'accord inter-annotateur en fonction du nombre de prémisses**. Valeur du coefficient de corrélation de Pearson : 0.021989. A droite une représentation graphique de **la complexité estimée moyenne en fonction du nombre de prémisses** Valeur du coefficient de corrélation de Pearson : 0.175165.

On observe que les corrélations entre le nombre de prémisses et l'accord ou entre le nombre de prémisses et la complexité sont faibles.

Conclusion : Les résultats ne permettent pas de confirmer les hypothèses 4 et 5 "Il y a corrélation entre le nombre de prémisses et l'accord inter-annotateur et entre le nombre de prémisses et la complexité" et tendraient à l'infirmer.

Un exemple qui pourrait illustrer cette absence de relation seraient des problèmes très simples décomposés en plusieurs prémisses :
Problème 18 :

P1 Tout Européen a le droit de vivre en Europe.
P2 Tout Européen est une personne.
P3 Toute personne qui a le droit de vivre en Europe peut circuler librement en Europe.
H Tout Européen peut circuler librement en Europe.
Rep Vrai

Et à l'inverse quelques problèmes difficiles à comprendre qui eux ne sont expliqués qu'en une seule phrase :
Problème 191 :

P Guillaume a suggéré au patron de Franck qu'ils devraient se rendre ensemble à la réunion et Charles (l'a suggéré) à la femme d'Alain..
H S'il a été suggéré que Guillaume et Franck devraient se rendre ensemble à la réunion, a-t-il été suggéré que Charles et Alain devraient se rendre à la réunion ensemble ?
Rep Vrai

Cela nous a donc amené à chercher d'autres facteurs de complexité et de désaccord.

Hypothèse 6 et 7 : Il y a corrélation entre la complexité syntaxique d'un problème et sa complexité estimée et entre la complexité syntaxique et l'accord inter-annotateur

Pour mesurer la complexité en terme de syntaxe d'une phrase on calcule son flux maximal. Schématiquement le flux, mesuré entre deux mots, correspond au nombre de relations syntaxiques en cours. Donc si l'on y pense d'une façon plus visuelle comme sur l'illustration ci-dessous cela correspond aux nombre de "flèches" que l'on croise si on dessine une ligne verticale entre deux mots.

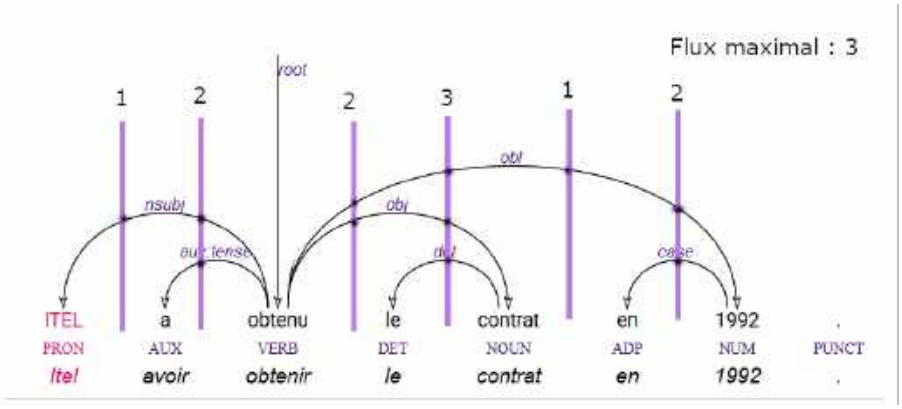

FIGURE 16 – Illustration du flux d'une phrase plutôt simple :"ITEL a obtenu le contrat en 1992"

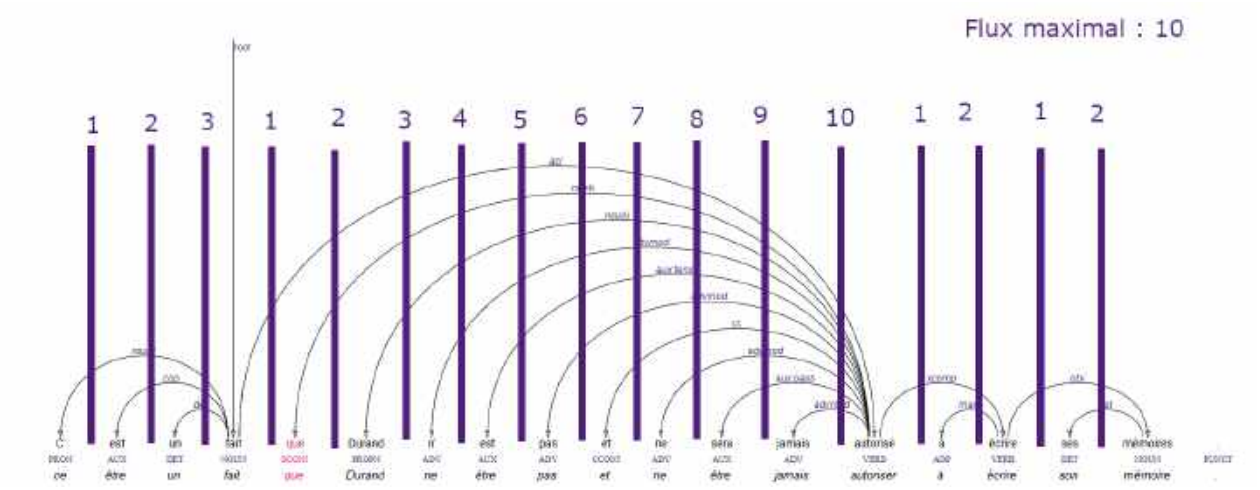

FIGURE 17 – Illustration du flux d'une phrase plus complexe : 'C'est un fait que Durand n'est pas et ne sera jamais autorisé à écrire ses mémoires.

Observons

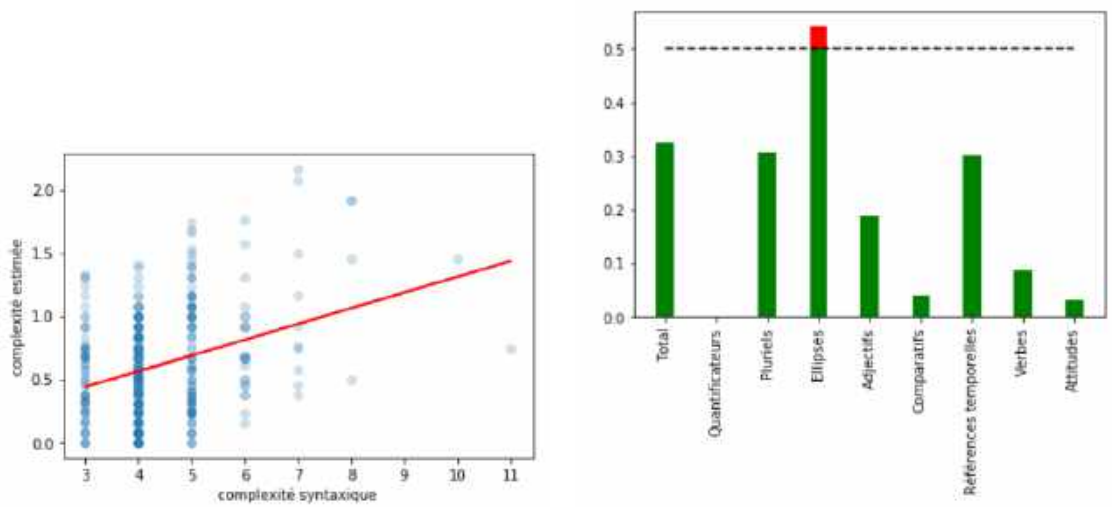

FIGURE 18 – A gauche une représentation graphique de **la complexité estimée en fonction de la complexité syntaxique** (flux maximal). A droite les valeur du coefficient de corrélation de Pearson selon les catégories de problèmes.

Ces résultats sont assez surprenant de prime abord puisqu'on ne constate qu'une faible corrélation entre la complexité syntaxique et la complexité estimée par les annotateurs hormis pour le cas des ellipses. Essayons de mieux comprendre ces résultats.

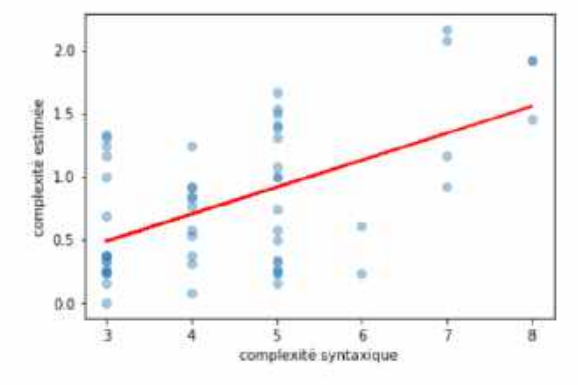

FIGURE 19 – Représentation graphique de **la complexité estimée en fonction de la complexité syntaxique** (flux maximal pour l'échantillon des problèmes comportant des ellipses)

La catégorie des ellipses comporte 55 problèmes, cependant étant donné la façon dont FraCaS est conçu on a, sur ce groupe, des inférences construites de manière très similaire qu'on peut regrouper finalement à une quinzaine de sous groupes de déclinaisons d'un même problème. Par exemple les problèmes 191 à 195 sont de légères variations du problème vu plus tôt "Guillaume a suggéré au patron de Franck qu'ils devraient se rendre ensemble à la réunion et Charles (l'a suggéré) à la femme d'Alain." Pour voir s'il y a vraiment une corrélation entre complexité syntaxique et estimée par les utilisateurs pour les ellipses il faudrait je pense un échantillon plus varié. On peut cependant essayer de réfléchir à ce qui ferait que le cas des ellipses serait particulièrement sensible à la complexité syntaxique.

Prenons l'exemple du problème 170 :

P Jean a trouvé Marie avant Guillaume.

Hyp Jean a trouvé Marie avant que Jean n'ait trouvé Guillaume.

Ici le flux maximal se trouve dans l'hypothèse et est de 7. La complexité estimée par les annotateurs moyenne est de 1.16 (soit 1.28 écart-type au dessus de la moyenne). Ce qu'on constate ici c'est que l'hypothèse est une interprétation de l'ellipse contenue dans la prémisse. La complexité de résolution de cette inférence réside principalement dans l'ambiguïté de sa prémisse puisqu'il est possible également de l'interpréter comme "Jean a trouvé Marie avant que Guillaume n'ait trouvé Marie". Admettons que l'hypothèse ait été "Guillaume a trouvé Marie après Jean" ou "Jean a trouvé Guillaume après Marie". Puisque ces phrases sont des expressions syntaxiquement plus simples de l'hypothèse mais avec un sens équivalent, le flux maximal de ce problème aurait été de 3 mais l'ambiguïté sur la prémisse serait restée. De plus si on retire la série de problèmes dérivés "Guillaume a suggéré au patron de Franck qu'ils devraient se rendre ensemble à la réunion et Charles (l'a suggéré) à la femme d'Alain." le coefficient de corrélation entre les complexités syntaxique et estimées par les annotateurs on passe de 0,54 à 0,21 ce qui remet en question cette éventuelle corrélation.

Conclusion : On ne peut pas confirmer que la complexité syntaxique a une influence sur la complexité ressentie par les annotateurs.

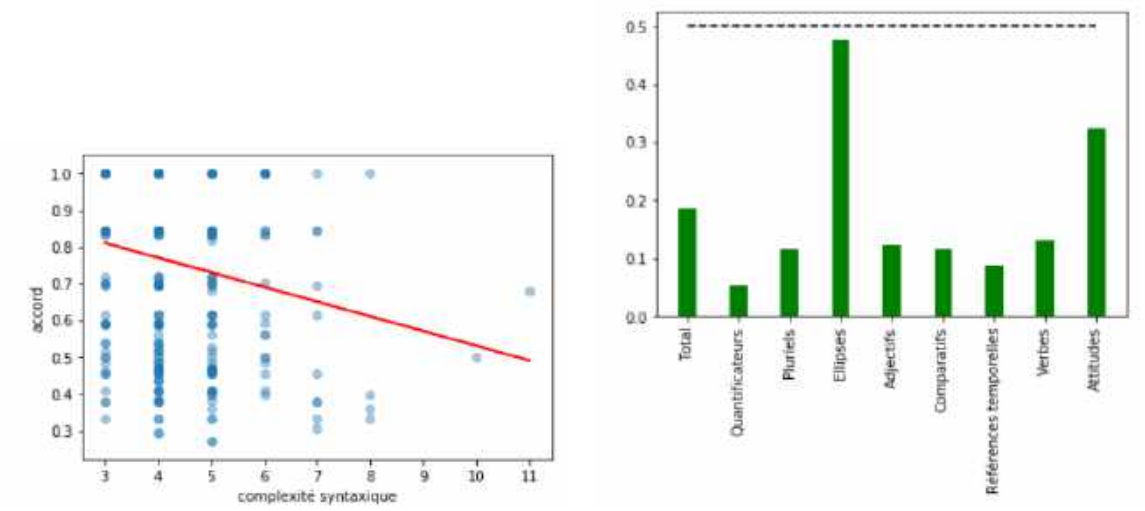

FIGURE 20 – A gauche une représentation graphique de **l'accord inter-annotateurs en fonction de la complexité syntaxique** (flux maximal). A droite les valeur du coefficient de corrélation de Pearson selon les catégories de problèmes

On ne remarque pas non plus de corrélation entre l'accord inter-annotateurs et la complexité syntaxique. **Conclusion** : On ne peut pas confirmer que la complexité syntaxique a une influence sur l'accord inter-annotateurs.

Conclusion sur l'analyse statistique des données issues de l'étude participative

Pour cette tâche au début je ne trouvais que peu de nouvelles pistes sur lesquelles travailler et apporter des informations intéressantes (hormis mettre à jour les analyses avec les nouvelles données récoltées). Mais ce qui est intéressant je trouve c'est qu'elle a finalement pris une place prépondérante dans mon stage et m'a amenée à beaucoup de réflexions intéressantes.

9 Proposer de nouveaux problèmes d'inférence

Enfin j'ai pu faire quelques propositions dans le but de participer à la future création d'une suite d'inférence pour le français inspirée de FraCaS. En effet un des problèmes actuels de la suite en français est son manque de naturel (beaucoup d'inférences autour du monde de l'entreprise, de clients etc) mais aussi le fait qu'il s'agit d'une traduction. Pourquoi est-ce un problème ? D'abord car il est des cas où des phénomènes et subtilités illustrés dans la version originale n'existent pas en français.

Exemple du Problème 116 :
Version originale
P Mary used **her** workstation.
H Mary is female.
Traduction
P Marie a utilisé **son** ordinateur.
H Marie est une femme.

Ce problème illustre en anglais l'inférence qu'on pourrait faire à partir l'adjectif possessif "her" sur le genre du possesseur (féminin). Cependant comme ce n'est pas possible en français avec "son", la traduction de cette inférence n'a donc pas vraiment de sens.

Ensuite car à l'inverse il serait pertinent de trouver et d'illustrer des phénomènes linguistiques propres au français. Par exemple pour rester sur un exemple portant sur le genre du sujet de la phrase on pourrait imaginer une inférence sur l'article telle que :

P La pilote a salué les passagers de l'avion
H : L'avion est piloté par une femme

Ici l'article est le seul moyen de déterminer le genre de la pilote (cela pourrait fonctionner pour la maire, la juge...) ce qui serait impossible dans la traduction anglaise "The pilot greeted the passengers on the plane". Pour trouver des phénomènes linguistique à étudier on m'a dirigé vers le **Handbook of French Semantics**. J'ai travaillé sur le premier chapitre portant sur les déterminants et ai essayé de décliner des inférences avec différents déterminants pour relever des différences selon les situations et des exemples intéressants.

Exemple 1 :
Déterminant : Un
P Un chat miaule
H Un chat roux miaule
— *Pas assez d'informations*
Déterminant : Tout/ Toute/ Tous / Toutes
P Tout chat miaule
H Tout chat roux miaule
— *Oui*

Exemple 2 :
Déterminant : Un
P Un étudiant de première année est riche
H Un étudiant est riche
— *Vrai*
Déterminant : Deux exactement / Exactement deux
P Exactement deux étudiants de première année sont riches
H Exactement deux étudiants sont riches
— *Pas assez d'informations*

Cinquième partie
Autres évènements auxquels j'ai assisté pendant le stage

10 Conférence JEP-TALN-RECITAL 2020

Il s'agit d'un ensemble de conférences et d'ateliers qui se sont tenus du 8 au 19 juin et portant sur le Traitement Automatique des Langues. Ils étaient organisés en ligne cette année par le CNRS et l'Université de Lorraine et plus particulièrement le LORIA, l'ATILF (Analyse et Traitement Informatique de la Langue Française) et l'institut INIST (Institut National de l'Information Scientifique et Technique).

Ils ont réuni des acteurs francophones internationaux pour présenter des travaux de recherches de ce domaine mais ayant des thématiques variées. J'ai pu par exemple assister à deux conférences, et les thématiques qui m'ont particulièrement intéressées étaient l'utilisation du TAL (et le recul, la réflexion qui était prise sur les attentes des acteurs mais aussi sur la réalité de ce qui avait pu être réalisé) ainsi qu'un travail portant sur les fautes à l'écrit de personnes atteintes de dyslexie dans le but de concevoir un outil d'aide à la rédaction adapté.

En parallèle de mon stage, c'était un panorama intéressant de l'étendue du domaine du TAL et de la Linguistique Informatique. Tandis que durant mon stage j'ai pu m'intéresser à des questions théoriques telles que celles de l'analyse des inférences ; grâce à ces conférences j'ai pu également avoir un aperçu d'applications très concrètes et utiles (comme le soutien de personnes atteintes de dyslexie dans leur vie de tous les jours).

11 Réunion avec l'équipe Sémagramme

J'ai pu également assister à une réunion de l'équipe Sémagramme en visioconférence ce qui était intéressant pour voir comment en ces conditions sanitaires particulières le travail de recherche et la collaboration qu'elle implique se poursuit. Comme avec mes tuteurs, ce sont des réunions hebdomadaires. On y fait un tour de table pour parler de l'avancement de chacun dans son travail, des

problèmes rencontrés, d'évènements à venir (conférences, publications d'article...). C'est l'occasion d'échanger et de demander parfois l'aide d'autres membres de l'équipe sur des questions où ils ont plus d'expertise. Par exemple dans le cadre du stage pour demander un retour sur les formulations logiques des problèmes.

Sixième partie

Outils informatiques et conceptuels utilisés

Ce stage commence dans des conditions un peu particulières puisque dans le contexte de la pandémie du Covid-19 nous devons adapter nos façons de travailler aux contraintes sanitaires. Autrement dit nous essayer au télétravail. J'ai donc travaillé sur mon ordinateur personnel. Mon système d'exploitation étant habituellement Windows j'ai dû utiliser une machine virtuelle pour pouvoir installer certains outils sous Linux (Grew et UDPipe). J'ai majoritairement travaillé en utilisant le langage Python et ses nombreuses librairies.

12 Grew [1]

Grew est un outil de réécriture de graphes (Graph Rewriting) pour le TAL développé au Loria. C'est-à-dire qu'il s'agit d'un outil. Il permet d'analyser (parsing) un énoncé en terme de syntaxe (Voir figure 21), de réaliser des représentation sémantiques et enfin de retrouver des motifs correspondant à un modèle ("match a pattern", Voir figure 25).

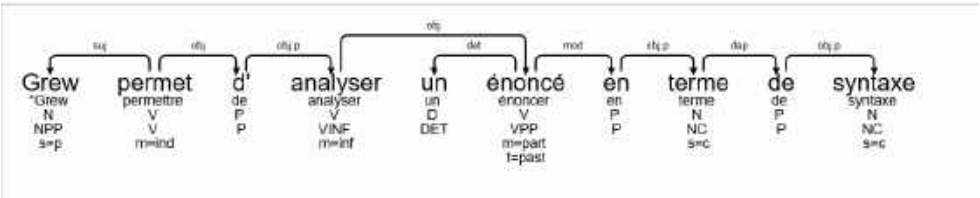

FIGURE 21 – Analyse syntaxique réalisée à l'aide de Grew Parse

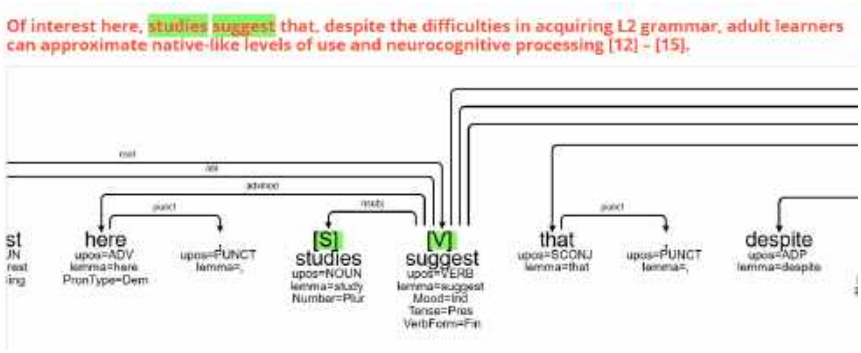

FIGURE 22 – Reconnaissance de pattern avec Grew Match, ici par ex : "S est sujet de V"

13 Universal Dependencies et UD Pipe

Universal Dependencies[16] est une norme et un ensemble de corpus annotés de manière homogène dans de nombreuses langues pour permettre un inventaire universel de catégories morphosyntaxiques et un étiquetage cohérent entre les langues. Cette recherche d'unification des annotations de dépendances a pour but d'avoir la possibilité de faire des parallèles entre les langues.

Les tokens (unités lexicales) de ces corpus sont les mots au sens syntaxique (donc une contraction comme au = à le en Français est considérée comme deux mots). A chaque mot sont associées des informations morphologiques (étiquette morphosyntaxique, genre, nombre relations, dépendances...)

Plus précisément durant ce stage j'ai utilisé le corpus UD_French-GSD [17] portant comme son nom l'indique sur le français et développé entre autre par des membres de l'équipe Sémagramme.

13.1 UDPipe[2]

UDPipe est un outil ("trainable pipeline") permettant la segmentation (en unités lexicales/-tokens), l'étiquetage, la lemmatisation (association au lemme, voir un peu plus loin) et l'analyse des dépendances (relations entre les mots) suivant les préconisations d'Universal Dependencies. Il produit des fichiers au format CONLL-U.

13.2 Format CoNLL-U[3]

Il s'agit d'un format de représentation des annotations où chaque ligne représente un mot et ses caractéristiques avec différentes informations :

- ID,
- FORM, : forme présente dans le texte
- LEMMA, : lemme sous forme "canonique" sans variation dues au genre et au nombre (flexions)
- UPOS (étiquette morpho-syntaxique universelle),
- XPOS (étiquette spécifique),
- FEATS (autres propriétés),
- HEAD (mot auquel se réfère le token),
- DEPREL (relation à ce mot),

— MISC (divers)

Exemple :

```
# sent\_id = 338H
# text = ITEL a obtenu le contrat en 1992 .
1 ITEL Itel PROPN _ Gender=Masc|Number=Sing 3 nsubj _ _
2 a avoir AUX _ Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin 3
aux:tense _ _
3 obtenu obtenir VERB _ Gender=Masc|Number=Sing|Tense=Past|VerbForm=Part 0
root _ _
4 le le DET _ Definite=Def|Gender=Masc|Number=Sing|PronType=Art 5 det _ _
5 contrat contrat NOUN _ Gender=Masc|Number=Sing 3 obj _ _
6 en en ADP _ _ 7 case _ _
7 1992 1992 NUM _ _ 3 obl _ SpaceAfter=No
8 . . PUNCT _ _ 3 punct _ _
```

Cela permet par exemple d'avoir des fichiers facilement exportables (simple fichier texte), éditables et analysable (librairies Python associées).

14 Arborator-Grew [4]

Il s'agit d'un outil d'analyse syntaxique collaboratif basés sur des fichiers CONLL-U. Il permet de proposer une annotation alternative à celle d'autres utilisateurs (sans supprimer l'originale). Voir fig. 4

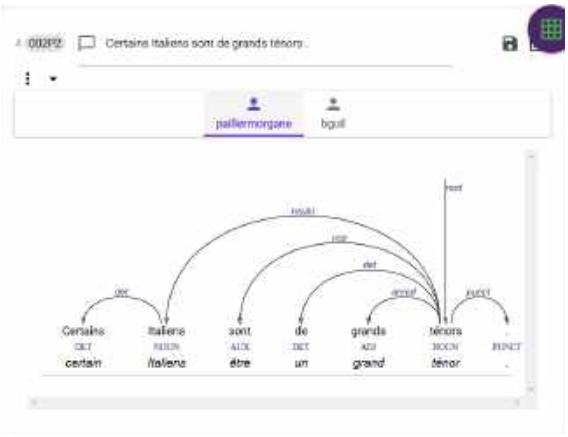

FIGURE 23 – Capture d'Arborator-Grew

Ces outils sont aussi associés aux fonctionnalités de Grew-match permettant par exemple de rechercher dans le corpus des relations syntaxiques, des n-grammes (suites de n mots), une forme d'un mot en particulier etc... en utilisant la syntaxe de Grew. Voir 25

FIGURE 24 – Capture de la recherche du bigramme "en train" dans notre corpus FraCaS sur Arborator

15 Etherpadlite[5]

Etherpadlite est un éditeur de texte collaboratif en ligne et libre (sur le même principe que l'éditeur plus connu Framapad).

16 Discord[6]

Discord est un outil de visioconférence et messagerie instantanée assez répandu. Il permet notamment le partage d'écran et propose des fonctionnalités intéressantes telles qu'un "filtre" des bruits parasites.

17 Overleaf[7]

Overleaf est un éditeur de texte en ligne permettant le travail à plusieurs sur un même document. Il a la particularité d'être basé sur le langage LaTeX et donc rendant possible d'incorporer des formules mathématiques et logiques, de mise en forme plutôt simple et modulable.

18 GitLab [8] [9]

GitLab est un logiciel de gestion de projet et une plateforme de développement open source et collaborative. Il permet comme d'autres applications similaires telles que GitHub de mettre en ligne des ressources et d'en gérer les versions. Il limite le risque de dégradation du travail accompli suite à une modification malencontreuse grâce à un système de validation et un suivi des propositions pour une tâche en cours.

19 Jupyter Notebook[10]

Il s'agit d'un outil permettant d'écrire, éxécuter et enregistrer des lignes de code et leur résultat (ici en Python) et également d'incorporer des titres et du texte. L'avantage par rapport à un environnement de programmation classique est la possibilité d'exécuter le script en entier ou simplement un extrait du code

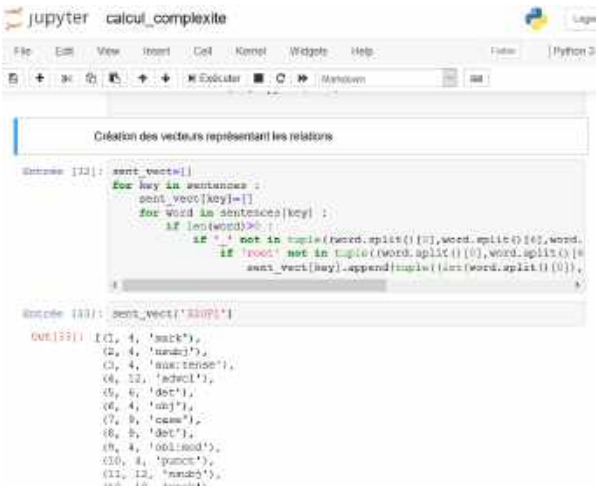

FIGURE 25 – Capture d'écran d'une utilisation de Jupyter Notebook où on peut voir du texte, du code et des résultats d'une exécution précédente sauvegardés

Septième partie
Impressions sur le stage

Cette expérience a été très positive. J'ai apprécié la variété des tâches que j'ai pu effectuer dans mon stage :

- Participer au crowdsourcing
- Proposer une expression sous forme logique des problèmes
- Procéder à l'étiquetage morphosyntaxique et l'analyse des dépendances de la ressource
- Répondre à une étude similaire et en étudier le rapport
- Reprendre les scripts d'analyse statistiques de ma prédécesseur et les approfondir
- Proposer de nouveaux problèmes d'inférence

J'ai vraiment senti que je pouvais être force de proposition et ce qui a pu être fait sortait du sujet de départ (notamment sur l'étude statistique).

J'ai pu également travailler ma capacité à actualiser mes connaissances par moi même vis à vis de Python, et des statistiques.

Du fait du télétravail, j'ai eu beaucoup de liberté dans mon organisation. Ce fut un élément à double tranchant qui a demandé une certaine discipline sur soi même pour que la travail avance. J'ai apprécié car je trouve que cela m'a responsabilisée. J'y ai également appris en terme de communication, gestion du temps et d'autonomie.

Cela m'a permis d'avoir un aperçu du domaine de la recherche et de ses pratiques ce qui m'a permis de réfléchir à mon projet professionnel, un de mes objectifs avec ce stage.

Ce stage a donc très bien répondu à mes attentes et j'espère avoir donné satisfaction à mes tuteurs dans les tâches qu'ils m'ont confié.

Références

[1] Grew. http://grew.fr/. Accessed : 2020-08-13.

[2] Milan Straka and Jana Straková. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[3] CoNLL-U Format. https://universaldependencies.org/format.html. Accessed : 2020-08-13.

[4] Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. When Collaborative Treebank Curation Meets Graph Grammars. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5293–5302, Marseille, France, May 2020. European Language Resources Association.

[5] Etherpadlite. https://etherpad.org/. Accessed : 2020-08-14.

[6] Discord. https://discord.com/new. Accessed : 2020-08-14.

[7] Overleaf. https://www.overleaf.com/. Accessed : 2020-08-14.

[8] Gitlab. https://about.gitlab.com/. Accessed : 2020-08-14.

[9] Gitlab : qu'est-ce que c'est ? https://junto.fr/blog/gitlab/. Accessed : 2020-08-14.

[10] Jupyter notebook. https://jupyter.org/. Accessed : 2020-08-14.

[11] Robin Cooper, Richard Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp, Manfred Pinkal, David Milward, Massimo Poesio, Stephen Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. Using the framework. Technical report, FraCaS : A Framework for Computational Semantics, 1996.

[12] Larousse, définition d'inference. https://www.larousse.fr/dictionnaires/francais/inf%C3%A9rence/42907. Accessed : 2020-08-13.

[13] Robin Cooper, Stergios Chatzikyriakidis, and Simon Dobnik. Testing the FraCas test suite. Presentation at the Unshared Task "Theory and System analysis with FraCaS, MultiFraCaS and JSeM Test Suites" of Logical Engineering of Natural Language Semantics 13 (LENLS 13), 2016.

[14] Maxime Amblard, Clément Beysson, Philippe de Groote, Bruno Guillaume, and Sylvain Pogodalla. A French version of the FraCaS test suite. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5887–5895, Marseille, France, May 2020. European Language Resources Association.

[15] Étude des préférences dans l'expression de la quantification. https://jeremieroux.fr/TERL3/images/Rapport%20TERL3.pdf. Accessed : 2020-08-24.

[16] Universal Dependencies. https://universaldependencies.org/. Accessed : 2020-08-13.

[17] UD_French-GSD. https://universaldependencies.org/treebanks/fr_gsd/. Accessed : 2020-08-13.

Huitième partie
Annexe

20 Types d'inférences

Dans la ressource initiale (en anglais) et dans sa traduction française, les problèmes sont répartis comme suit :

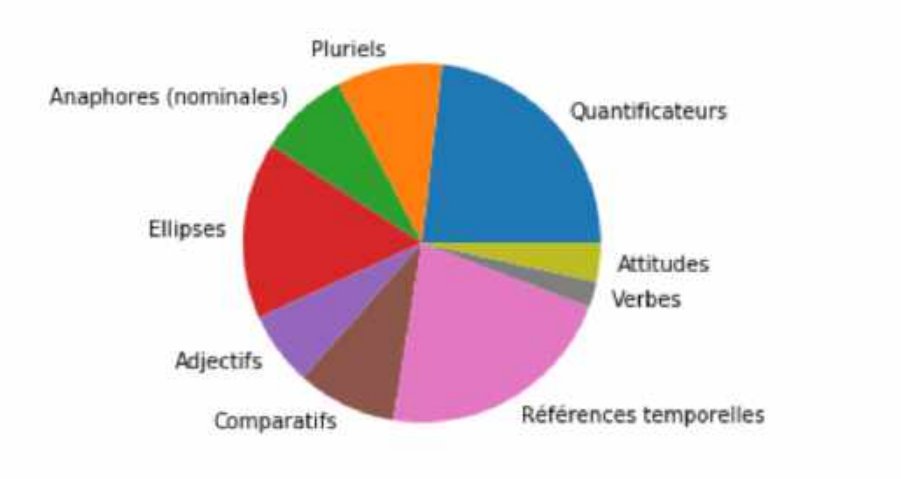

FIGURE 26 – Répartition des types de problèmes

— 80 inférences (23%) sont basées sur différents emplois de **quantificateurs** (dans le langage naturel donc ici "all", "some", "each","there is/are"....)

A Scandinavian won a Nobel prize. Every Swede is a Scandinavian.
Did a Swede win a Nobel prize ?

[Dont'know]

— 33 inférences (10%) sont basées sur des **pluriels** (ex : plusieurs noms dans le sujet et le problème portant sur savoir qui dans ce groupe a fait l'action, pluriels se comportant comme des quantificateurs existentiels ou quasi universels, ou comme des généralités, comme représentant un collectif mais pas forcément tous ses individus)

The inhabitants of Cambridge voted for a Labour MP
Did every inhabitants of Cambridge voted for a Labour MP ?

[Dont'know]

— 28 inférences (8%) sont basées sur des **anaphores nominales** : dans la phrase avec des pronoms possessifs, entre les phrases avec des pronoms relatifs, "pronoms d'ânes" (Donkey pronouns, soit des pronoms liés sémantiquement mais pas syntaxiquement, comme dans la phrase "Tout fermier qui possède un âne le bat"), avec des pronoms réfléchis

John said Bill had hurt himself
Did John say Bill had been hurt ?

[Yes]

— 55 inférences (16%) sont basées sur des **ellipses** c'est-à-dire que les conclusions sont basées sur des éléments éludés dans les déclarations, avec des structures comme une deuxième déclaration reprenant implicitement des éléments de la première phrase notamment là encore avec des anaphores

$$\frac{\text{John owns a car. Bill owns one too.}}{\text{Does Bill own a car ?}}$$

[Yes]

— 23 inférences (7%) sont basées sur des **adjectifs** influant ou non sur l'existence, la nature du nom auxquels ils se réfèrent (par exemple utiliser un"gros"diamant est un diamant mais un "faux" diamant n'en est pas un donc ici l'adjectif a une grande influence sur le sens de la phrase). On a aussi des adjectifs permettant de comparer des éléments entre eux comme "grand", "petit" (qui peuvent dans certains cas entrer en contradiction avec le noms auxquels ils se réfèrent si ceux ci entrent dans des classes, par exemple : Un éléphant est un gros animal donc un petit éléphant reste un gros animal).

$$\frac{\text{John is a former successful university student}}{\text{Is John successful ?}}$$

[Dont'know]

— 31 inférences (9%) sont basées sur des **comparaisons**

$$\frac{\text{ITEL sold 3000 more computers than APCOM.APCOM sold exactly 2500 computers}}{\text{Did ITEL sell 5500 computers ?}}$$

[Dont'know]

— 75 inférences (22%) sont basées sur des **références temporelles** : utilisation des temps (conjugaison au passé, futur) adverbes ("before", "after"), prépositions ("in", "for"), associations avec les quantificateurs ("every") avec une difficulté liée à la nature des éléments décrits (un accomplissement par exemple ne se répète pas, quelqu'un qui commence une activité plus tôt qu'un autre ne la finira pas forcément plus tôt...)

$$\frac{\text{Smith wrote his first novel in 1991}}{\text{Did Smith write his first novel in 1992 ?}}$$

(Unrepeatable) accomplishment[No]

— 8 inférences (2%) sont basées sur différents aspects des **verbes** par exemple des nuances liées aux formes de présent continu (be + -ing)

$$\frac{\text{ITEL was winning the contract from APCOM in 1993}}{\text{Did ITEL win a contract in 1993 ?}}$$

[Don't know]

et savoir si le verbe en tant que prédicat se distribue à plusieurs sujets

$$\frac{\text{Smith and Jones left the meeting.}}{\text{Did Smith leave the meeting ?}}$$

[Yes]

— 13 inférences (4%) sont dites d'"attitudes" basées sur des structures telles que A knew that / managed to B $\rightarrow$ B ou A said/believed/denied... B $\not\rightarrow$ B, ou alors basées sur le verbe "see" qui sert parfois selon les cas de preuve. (" $a\,saw\,\phi \rightarrow \phi$")

$$\frac{\text{Smith saw Jones sign the contract and his secretary make a copy}}{\text{Did Smith see Jones sign the contract ?}}$$

[Yes]

21 Propositions d'expressions logiques pour les inférences

Ici un exemple sera détaillé par type d'inférences

Quantificateurs

Problème 10, Conservativité
P La plupart des grands ténors sont italiens.
Hyp Il y a de grands ténors qui sont italiens.
Rep Vrai
Formulation logique :
— L_T : La plupart
— $grand\_tenor(x)$
— $italien(x)$
— P : $L_T\, x, grandTenor(x) \rightarrow italien(x)$
— Hyp : $\exists x, grandTenor(x) \wedge italien(x)$

Comme dit plus haut un ensemble de problèmes se sont portés sur des inférences liées aux quantificateurs. Un des questionnements a donc été de pourvoir choisir comment les modéliser soit en utilisant des quantificateurs classiques des mathématiques et de la logique (pour tout, il existe) soit en en créant des spécifiques. Nous avons effectué une réunion pour revenir avec d'autres membres de l'équipe Sémagramme ayant aussi travaillé sur FraCaS, Philippe de Groote et Sylvain Pogodalla, afin de retravailler les expressions logiques que nous avions produites. Par exemple ici pour le cas du quantificateur "La Plupart" ("Most") L'idée ici était de donner la sémantique de ce quantificateur plutôt que comme nous l'avions fait plus haut le nommer sans le définir. Ici le quantificateur fonctionne sur des ensembles, l'idée étant que si l'intersection de l'ensemble des Grands Ténors et de l'ensemble des Italiens est supérieur à la moitié des Grands Ténors alors on a La Plupart des Grands Ténors qui sont Italiens.

— G : ensemble des Grands ténors
— I : ensemble des Italiens

Ou alors

— sémantique de MOST x (A x)(B x) : $2x|A \wedge B| > |A|$
— P : $MOSTx(G\,x)(I\,x)$
— Hyp : $\exists x, x \in G \cap I$

Logique Modale

Problème 20 Monotonicité vers le haut sur le second argument

P1 Tout Européen a le droit de vivre en Europe.
P2 Tout Européen est une personne.
P3 Toute personne qui a le droit de vivre en Europe peut circuler librement en Europe.
Hyp Tout Européen peut circuler librement en Europe.
Rep Vrai

Formulation logique :

— $europeen(x)$
— $personne(x)$
— $vivre\_europe(x)$
— $circuler\_librement\_europe(e)$
P1 $\forall x \exists e.europeen(x) \rightarrow vivre\_europe(e) \wedge \Diamond Agent(e, x)$
P2 $\forall x.europeen(x) \rightarrow personne(x)$
P3 $\forall x \exists e \exists e'.personne(x) \wedge vivre\_europe(e) \wedge \Diamond Agent(e, x)) \rightarrow (circuler\_librement\_europe(e') \wedge \Diamond Agent(e', x)$
Hyp $\forall x \exists e.europeen(x) \rightarrow circuler\_librement\_europe(e) \wedge \Diamond Agent(e, x)$

Ici pour modéliser que pour chaque Européen (x) le prédicat $circuler\_librement\_europe(e) \wedge Agent(e, x)$ est satisfiable mais pas toujours vrai, nous avions donc choisi d'introduire la modalité "possible" représentée par le symbole diamant($\Diamond$).

Cependant à nouveau durant cette réunion avec d'autres membres de Sémagramme il a paru plus pertinent d'une certaine façon de simplifier la notation n'utilisant pas les opérateurs de modalité mais en remplaçant par exemple "vivre_europe" par "droit_vivre_europe" .

— $europeen(x)$
— $personne(x)$
— $droit\_vivre\_europe(x)$
— $pouvoir\_circuler\_librement\_europe(x)$
P1 $\forall x \exists e.europeen(x) \rightarrow droit\_vivre\_europe(e) \wedge Agent(e, x)$
P2 $\forall x.europeen(x) \rightarrow personne(x)$
P3 $\forall x \exists e \exists e'.personne(x) \wedge droit\_vivre\_europe(e) \wedge Agent(e, x)) \rightarrow (pouvoir\_circuler\_librement\_europe(e') \wedge Agent(e', x)$
Hyp $\forall x \exists e.europeen(x) \rightarrow pouvoir\_circuler\_librement\_europe(e) \wedge Agent(e, x)$

Introduction d'évènements

Dans le cas de descriptions d'actions il a paru pertinent d'introduire des variables d'évènements dans les expressions logiques

Problème 140, anaphore reflexive simple

P Jean a dit que Guillaume s'était blessé.

H Jean a dit que Guillaume avait été blessé.

Formulation logique :

— JEAN

— GUILLAUME

— $blesser(e)$: e est l'action de blesser

— $dire(x, y)$: x a dit y

P1 $\exists e \exists e'.blesser(e) \wedge Agent(e, \text{GUILLAUME}) \wedge Patient(e, \text{GUILLAUME}) \wedge dire(e') \wedge Objet(e', e) \wedge Agent(e', \text{JEAN})$

H $\exists e \exists e'.blesser(e) \wedge Patient(e, \text{GUILLAUME}) \wedge dire(e') \wedge Objet(e', e) \wedge Agent(e', \text{JEAN})$

Ici par exemple l'usage d'évènements se justifie par la nuance entre a)"Guillaume s'[est] blessé" et b)"Guillaume [a] été blessé" avec dans le cas a) une information sur l'Agent de l'action mais pas dans le cas b). De plus ici l'action "dire" rapporte l'évènement "Guillaume s'[est] blessé"/"Guillaume [a] été blessé" il fallait donc une variable pour représenter cet évènement et aussi un prédicat pour lier l'action $dire(e)$ et l'évènement $blesser(e')$: $Objet(e', e)$

Variables temporelles

Il s'agissait ici de pouvoir nuancer la valeur de vérité d'un prédicat en ne la rendant vraie qu'à une période particulière.

Problème 200, Adjectifs affirmatifs et non affirmatifs

P Jean est un ancien remarquable étudiant.

H Jean est remarquable.

Formulation logique :

— JEAN : Jean

— $etudiant(x)$ x est étudiant à t

— $passe(t)$: t appartient au passé

— MAINTENANT

— $remarquable(x, t)$: x est remarquable à un temps t

— $remarquable(x, t)$: x est remarquable à un temps t

P $\exists t.passe(t) \wedge etudiant(\text{JEAN}, t) \wedge remarquable(\text{JEAN}, t)$

H $\exists t.(t = \text{MAINTENANT}) \wedge remarquable(\text{JEAN})$

Mais il s'agissait aussi de pouvoir comparer les durées d'une action comme ici :

Problème 30 Monotonicité vers le haut sur le second argument

P Aucun des deux commissaires ne passe beaucoup de temps à la maison.

Hyp Aucun des deux commissaires ne passe du temps à la maison.

Rep Pas assez d'informations

Formulation logique :

— $commissaire(x)$

— e : évenement

— $passe\_temps\_maison(e)$: e passe du temps à la maison

— $duree(e,t)$: e dure t

— $magn(t)$: t représente beaucoup de temps

— C : Ensemble de deux commissaires (issus du contexte de la phrase)

P $\exists X \exists e \exists t.Card(X) = 2 \wedge (\forall x \in X \rightarrow commissaire(x)) \wedge (\forall x \in X.passe\_temps\_maison(e) \wedge Agent(e,x) \wedge duree(e,t) \wedge \neg magn(t))$

H $\exists X \exists e \exists t.Card(X) = 2 \wedge (\forall x \in X \rightarrow commissaire(x)) \wedge (\forall x \in X.\neg passe\_temps\_maison(e) \wedge Agent(e,x))$

Cependant durant la réunion avec d'autres membres de FraCaS plusieurs idées pour représenter ce problèmes ont été proposées

"Négation(Un commissaire passe beaucoup de temps à la maison)"

P $\exists X \exists e \exists t.Card(X) = 2 \wedge (\forall x \in X \rightarrow commissaire(x)) \wedge \neg(\exists x \in X.passe\_temps\_maison(e) \wedge Agent(e,x) \wedge duree(e,t) \wedge magn(t))$

H $\exists X \exists e \exists t.Card(X) = 2 \wedge (\forall x \in X \rightarrow commissaire(x)) \wedge \neg(\exists x \in X.passe\_temps\_maison(e) \wedge Agent(e,x))$

Finalement

P $\exists X.|X| = 2 \wedge (\forall x \exists t.x \in X \rightarrow (commissaire(x) \wedge \neg(passe\_temps\_maison(x,t) \wedge magn(t))$

H $\exists X.|X| = 2 \wedge (\forall x \exists t.x \in X \rightarrow (commissaire(x) \wedge \neg(passe\_temps\_maison(x,t)))$

Ensembles

Problème 80, Monotonicité vers le bas sur le premier argument

P Au plus dix commissaires passent du temps à la maison.

Hyp Au plus dix femmes commissaires passent du temps à la maison.

Rep Vrai

Formulation logique :

— $commissaire(x)$

— $femme(x)$

— e : évenement

— $passe\_temps\_maison(e)$: e passe du temps à la maison

— $duree(e,t)$: e dure t

— $magn(t)$: t représente beaucoup de temps

P $\exists e \exists E \, \forall x. Card(E) <= 10 \wedge (commissaire(x) \wedge passe\_temps\_maison(e)) \wedge Agent(e,x) \rightarrow x \in E$

Hyp $\exists e \exists E \, \forall x. Card(E) <= 10 \wedge (commissaire(x) \wedge femme(x) \wedge passe\_temps\_maison(e)) \wedge Agent(e,x) \rightarrow x \in E$

Adjectifs (et leur portée)

Problème 200

P Jean est un ancien remarquable étudiant.

H Jean est remarquable.

Formulation logique :

— JEAN : Jean

— $etudiant(x)$ x est étudiant à t

— $passe(t)$: t appartient au passé

— MAINTENANT

— $remarquable(x,t)$: x est remarquable à un temps t

P $\exists t. passe(t) \wedge etudiant(\text{JEAN}, t) \wedge remarquable(\text{JEAN}, t)$

H $\exists t. remarquable(\text{JEAN}, \text{MAINTENANT})$

Des problèmes liés à modélisations de certaines inférences m'ont amenées à lire l'article d'Ora Matushansky "Les adjectifs – Une introduction". **Intersectivité des adjectifs** Prenons l'exemple de l'article : "planète rouge". On dit d'un adjectif, ici "rouge", qu'il est intersectif si en pensant en termes d'ensembles, à l'intersection de l'ensemble des "planètes" et l'ensemble des objets "rouge" on trouve l'ensemble des "planètes rouges". Prenons le contre exemple "ancien étudiant" : On ne peut pas dire que toutes les occurrences d'"anciens étudiants" appartiennent à l'ensemble des "étudiants" ni des choses "anciennes". Il peuvent aussi être représentés comme satisfaisant la formule "$\exists P \forall Q \forall x [A](Q)(x) \equiv P(x)Q(x)$" "$\exists P \forall Q \forall x [rouge](planete)(x) \equiv rouge(x)planete(x)$" Statisfiable "$\exists P \forall Q \forall x [ancien](etudiant)(x) \equiv ancien(x)etudiant(x)$" Faux.

On distingue ici donc les adjectifs :

— intersectifs : "$\forall x. [italien](tenor)(x) \equiv italien(x)tenor(x)$"

— subsectifs : $\forall x.[habile](politicienne)(x) \rightarrow politicienne(x)$
— non subectifs simple : ancien étudiant ?
— non subsectif privatif : $\forall x.[faux](diamant)(x) \rightarrow diamant(x)$

Scalarité des adjectifs

"Le monarque est un grand papillon. → Le monarque est grand." Adjectif intersectifs pouvant être décrits comme vague, imprécis dont l'interprétation dépend du contexte. On trouve par exemple dans FraCaS plusieurs exemples du type :

Problème 210, Classes de comparaison extensives
P1 Toutes les souris sont de petits animaux.
P2 Mickey est une grande souris.
H Mickey est un grand animal.
Formulation logique :
— MICKEY
— $souris(x)$
— $animal(x)$
— $taille(x, y)$ x est la taille de y
— $inferieur(x, y)$ x est inférieur à y
— $superieur(x, y)$ x est supérieur à y
— L_T : la plupart

Petits animaux : animaux qui sont plus petits que la plupart des animaux Grande souris : souris qui est plus grande que la plupart des souris <=> la plupart des souris sont plus petites que cette souris Grand animal : animal qui est plus grand que la plupart des animaux <=> la plupart des animaux sont plus petits que cet animal