

Bonjour à toutes et à tous, je suis Camille Challant, je suis en deuxième année de thèse au LISN à Orsay et je travaille avec Michael Filhol. Je suis ravie de vous parler aujourd'hui de notre présentation qui s'intitule « de l'informatique à la linguistique, une grammaire pour les langues des signes basée sur AZee ».

## 1

Pour commencer, nous nous intéressons à la modélisation des langues des signes, qui est essentielle pour le traitement automatique, dans l'optique de faire les mêmes choses qu'avec les langues vocales : de la synthèse, reconnaissance voire même assister la traduction avec de la traduction automatique. Au laboratoire, nous travaillons sur un modèle qui s'appelle AZee, et qui est le modèle le plus avancé en matière de synthèse par avatar, avec l'avatar Paula de l'université DePaul à Chicago que l'on peut voir sur la slide. AZee a donc un objectif principalement informatique à la base. Mais en tant que linguiste, je vais essayer de vous montrer en quoi AZee se révèle être très pertinent d'un point de vue linguistique. Je vais d'abord commencer par vous expliquer le fonctionnement d'AZee dans les grandes lignes.

## 2

AZee est un modèle formel, au sens 'non-ambigu, qui puisse être lu par un ordinateur' fondé sur la notion de règle de production. C'est une fonction qui représente un sens interprétable et qui détermine un ensemble de formes à produire, qu'on trouve en étudiant des corpus de langue des signes.

Par exemple, en LSF, le sens 'chat' est toujours associé à la forme qui est représentée sous forme de diagramme. On voit la ligne horizontale qui représente le temps, cette première boîte spécifie la première position du signe, et plus loin dans le temps la deuxième position du signe. C'est une règle de production, qu'on appelle *chat*. Ensuite, un autre exemple, c'est la règle nommée *inter-subjectivity*. Le sens c'est « il est généralement admis que sig ». Cette règle a donc un argument, sig, qu'on représente par une boîte rouge qu'il va falloir remplir par n'importe quel énoncé. Se superpose à cet énoncé une posture des lèvres comme on le voit ici. On a aussi la règle *info-about* : « une information info, donnée à propos d'un topic » et on va avoir topic qui est maintenu, une pause dont la durée est contrôlée puis info qui est maintenu plus longtemps que pour le topic, et un clignement des yeux qui se superpose à ce maintien.

L'avantage de n'être basé que sur cette notion de règle de production et d'association entre forme et sens, c'est qu'on ne présuppose pas l'existence de niveaux linguistiques, d'une séquence, ou encore de catégories syntaxiques

## 3

Ces règles de production peuvent s'imbriquer entre elles, et permettent de former une expression qui représente la sémantique d'un énoncé en langue des signes, et qui génère les formes observées. On appelle ça des expressions AZee de discours.

Par exemple, on a combiné le sens des règles que je viens de présenter, cela donne l'expression que l'on peut voir sur la slide. On a donc *info-about* qui a pour topic *chat* et pour info *inter-subjectivity*, qui a elle-même pour argument *mignon*. J'ai combiné les règles grâce au sens, que l'on interprète comme « les chats sont généralement admis comme mignons » et cela donne ce diagramme pour ce qui est des formes à produire. On

reconnait encore ici la forme de *chat*, de *mignon* qui sont venues remplir nos boîtes rouges, mais aussi de *inter-subjectivity* ou *info about* qui sont toutes combinées entre elles.

C'est un exemple construit ici, on a donc commencé par se demander si l'on pouvait représenter des vrais discours entiers en LSF avec AZee.

#### 4

C'est donc ce que l'on a tenté de faire avec le corpus des 40 brèves, qui est un corpus d'une heure de LSF. On a 40 brèves journalistiques en français écrit, qui ont chacune été traduites en LSF par 3 traducteurs Sourds. On a totalement AZeefié ce corpus, c'est-à-dire que pour chacune des 120 vidéos, on a retrouvé l'expression AZee qui lui correspondait, en observant les formes et en interprétant le sens. En résultent 120 expressions AZee de discours (cf. extrait d'une des brèves à droite). Nous avons obtenu une couverture du corpus par le modèle de plus de 94 %. Tout a été publié à LREC en 2022 et le corpus est disponible sur Ortolang si cela vous intéresse. On a donc un corpus qui représente 1h de LSF, qui pouvait nous servir de base pour d'autres études.

#### 5

J'en arrive maintenant à ma question de recherche, à ce qui m'intéresse principalement dans ma thèse : « est-ce qu'il existe des contraintes qui régissent la construction de ces expressions ? »

Par exemple, si l'on prend 3 règles de production : *nb-1* et *nb-2* qui produisent respectivement ces formes et la règle *tens-units* avec deux arguments, qui permet de mettre deux blocs en séquence avec un bloc plus à droite que le précédent, que l'on va interpréter comme un nombre à 2 chiffres.

Avec ces 3 règles, 4 combinaisons sont théoriquement possibles : 1-1, 1-2, 2-1, 2-2

Maintenant, si on regarde dans des corpus de LSF, on se rend compte qu'on observe les deux dernières constructions (2-1 et 2-2), mais on ne les rencontre pas pour 1-1 et 1-2.

Une hypothèse pourrait être que la règle *tens-units* n'a jamais *nb-1* comme premier argument. C'est donc ce genre de contraintes qu'on aimerait faire émerger et trouver pour AZee. La question maintenant c'est « comment faire pour trouver ces contraintes ? »

#### 6

Pour *tens-units*, on vient de s'intéresser aux noms des règles impliquées dans l'expression, mais il y a d'autres tests possibles pour essayer de faire émerger des contraintes.

Par exemple, étant donné une expression, on va pouvoir tester le nom de la règle de production utilisée qui est à la racine de l'expression. Ici, c'est *info-about*. On va aussi avoir *argcount* pour compter le nombre d'arguments d'une règle de production : par exemple *info-about* a deux arguments (*topic* et *info*), *inter-subjectivity* 1 (*sig*) et *chat* et *mignon* n'en ont pas. Enfin, *prodcount* va compter le nombre de règle de productions contenues dans une expression (si on considère toute l'expression on a un *prodcount* de 4 ici : *info-about*, *chat*, *inter-subjectivity* et *mignon*).

Tous les tests sont combinables entre eux pour former des tests plus complexes. Par exemple, on peut chercher toutes les règles qui sont nommées *info-about* et qui ont un *prodcount* de 4.

On a donc pu utiliser ces requêtes sur le corpus d'expressions AZee qu'on venait de créer, qui contient environ 11 500 applications de règles : ça nous semblait être un bon point de départ pour de premières analyses.

## 7

Voilà un exemple pour illustrer notre méthode. On commence par se poser une question, comme ici «est-ce que tous les arguments des règles de production ont la même complexité ? »

Ensuite, pour répondre à cette question, on a utilisé le test « name » pour récupérer les expressions ayant pour racine telle règle suivie de tel arg, combiné avec « prodcount », qui compte le nombre d'applications de règle et qui reflète la complexité de l'expression.

C'est à dire, que par exemple j'ai cherché toutes les expressions qui avaient *info-about\_info* en racine, qui ont un prodcount de 1. Puis celles qui ont un prodcount de 2, de 3 etc. Et voici les distributions que l'on obtient. On se rend compte que l'on a affaire à des distributions très différentes.

## 8

Je vais maintenant vous présenter les observations que l'on a faites sur deux cas en particulier, qui présentent des contrastes intéressants. J'ai choisi la règle *category*, qui a deux arguments, où l'un est à interpréter comme une instance de l'autre. Par exemple : *category*(mois, poisson), qu'on interprète comme « avril ». Dans notre corpus, on a 500 applications de cette règle.

J'ai cherché le prodcount des expressions qui ont *category* en racine, mais aussi ceux qui ont *category\_cat* et *category\_elt*. On obtient ces distributions. Pour la règle prise dans son ensemble, le prodcount est compris entre 3 et 39, mais en réalité on voit qu'il y a comme une cassure vers 20. Pour *cat*, cette limite est encore plus petite (entre 1 et 5), alors qu'*elt* la limite est plus grande : les deux arguments ne semblent pas se comporter de la même manière, l'un est plus complexe que l'autre.

Enfin, si l'on s'intéresse aux règles contenues dans *category*, on se rend compte que ça varie selon l'argument. *Elt* ne contient jamais *prise-de-parole*, *context*, *nicht-sondern*. C'est pareil pour *cat*, mais on a 6 règles de plus que l'on n'observe jamais dans l'argument *cat*, qui semble beaucoup plus contraint.

## 9

Je peux faire la même chose pour *prise-de-parole*. C'est une règle qui a pour sens de rassembler ses idées et faire une petite pause avant de signer à nouveau. On trouve 165 applications de cette règle dans notre corpus.

Son prodcount va de 7 jusqu'à 126. C'est déjà très différent de ce qu'on observe avec *category*, où on avait énormément de prodcount faible, et plus le prodcount augmente moins il y a d'occurrences. Ici, il n'y a pas de *prise-de-parole* avec un prodcount très petit, il semble y avoir une limite minimale.

Enfin, cette règle peut contenir n'importe quelle règle (là où *category* a quelques restrictions) mais elle n'apparaît que dans les arguments de quelques règles comme *context*, *info-about* ou *each-of*.

## 10

D'après toutes ces observations, nous pouvons émettre quelques hypothèses : *category* a une limite maximale concernant le nombre de règles de production appliquées dans l'expression dont elle est la racine (son *prodcount*) alors que *prise-de-parole* a une limite minimale. Aussi, *category* peut apparaître dans n'importe quelle position, être contenue dans n'importe quelle règle de production alors que ce n'est pas le cas de *prise-de-parole*. Enfin, certaines règles ne sont jamais contenues dans *category*, ce qui n'est pas le cas pour *prise-de-parole*.

Ce qui nous intéresse également, c'est la nature de ces contraintes.

Si on accepte qu'une expression AZee est une représentation de la LSF, alors les contraintes qui existent dans la langue devraient se refléter dans les expressions. On considère donc que l'on a affaire à une contrainte grammaticale. Lorsqu'on les met ensemble, ces contraintes peuvent être perçues comme un système, et un système de contraintes grammaticales, c'est une grammaire formelle. On a donc envie de proposer une grammaire formelle pour la LSF dans un premier temps, basée sur AZee

## 11

On peut ensuite placer cette grammaire au sein du paysage des grammaires existantes. Notre grammaire AZee aurait plusieurs caractéristiques principales : la première c'est qu'il y a plusieurs choses qu'on ne présuppose pas : la séquence par exemple. On ne présuppose pas non plus de niveaux linguistiques traditionnels, ni de catégories syntaxiques. On ne parle jamais de verbe, de nom, d'adjectifs.. et on s'en sort sans, pour le moment. En fait, on revient à quelque chose de plus fondamental ! Ça ne veut pas dire qu'on nie l'existence de ces notions. Si ces notions se trouvent être pertinentes pour l'étude des LS, on pourra les redéfinir grâce à des critères formels ; mais ces notions n'étaient pas dans nos présupposés de base.

Une caractéristique importante également, c'est que dans nos expressions, chaque nœud est porteur de sens, puisque qu'une règle de production est une association forme-sens par définition. Enfin, nos expressions déterminent des formes à produire, articulateurs non-manuels inclus.

Au niveau du placement théorique avec les autres grammaires.. Tout d'abord, cette grammaire s'oppose aux grammaires génératives, centrées sur la syntaxe et basées sur la séquence, qui ne prennent pas en compte le sens : des nœuds type NP, VP sont un niveau d'abstraction comme on peut le voir dans l'exemple que j'ai emprunté à Fabian Bros et ne sont pas porteurs de sens, contrairement aux nôtres.

En revanche, plusieurs idées appartenant à d'autres courants nous intéressent. Par exemple, les grammaires cognitives accordent une grande place à la sémantique et cela nous paraît pertinent. On peut également apercevoir quelques points communs entre notre grammaire et les grammaires de construction, qui prennent également pour base l'association forme-sens présente dans les langues. Il y a également un continuum lexique-syntaxe et pas de niveau linguistiques clairement distingués. En ce qui concerne les grammaires de propriétés, c'est une approche qui nous intéresse beaucoup également : les propriétés sont des contraintes, on est sur un système de contraintes qui va permettre de caractériser les énoncés grâce à un continuum de grammaticalité et non de simplement faire une opposition grammatical/agrammatical.

Par contre, toutes ces grammaires utilisent toujours la notion de catégories syntaxiques.

## 12

Pour finir, nous avons plusieurs perspectives concernant notre travail.

Nous avons conscience qu'on travaille sur un corpus d'un genre particulier (le genre journalistique), et même si on réussit à observer des choses intéressantes, on a envie d'élargir notre champ d'étude. Par exemple avec un corpus qui contient de nombreuses structures iconiques comme Mocap 1.

On aimerait également dans le futur pouvoir générer des énoncés avec Paula (puisque c'est possible avec AZee) pour confirmer ou infirmer nos hypothèses. Est-ce qu'un locuteur natif accepterait certains énoncés qui ne répondraient pas à nos contraintes ? Par exemple, générer un énoncé avec un *prise-de-parole* dans un *category* (ce qui n'arrive jamais, comme on l'a vu) et voir la réaction des locuteurs.