Étude d'un corpus de requêtes en langue naturelle pour des agents assistants

François Bouchet
Jean-Paul Sansonnet

LIMSI-CNRS

{bouchet, jps}@limsi.fr

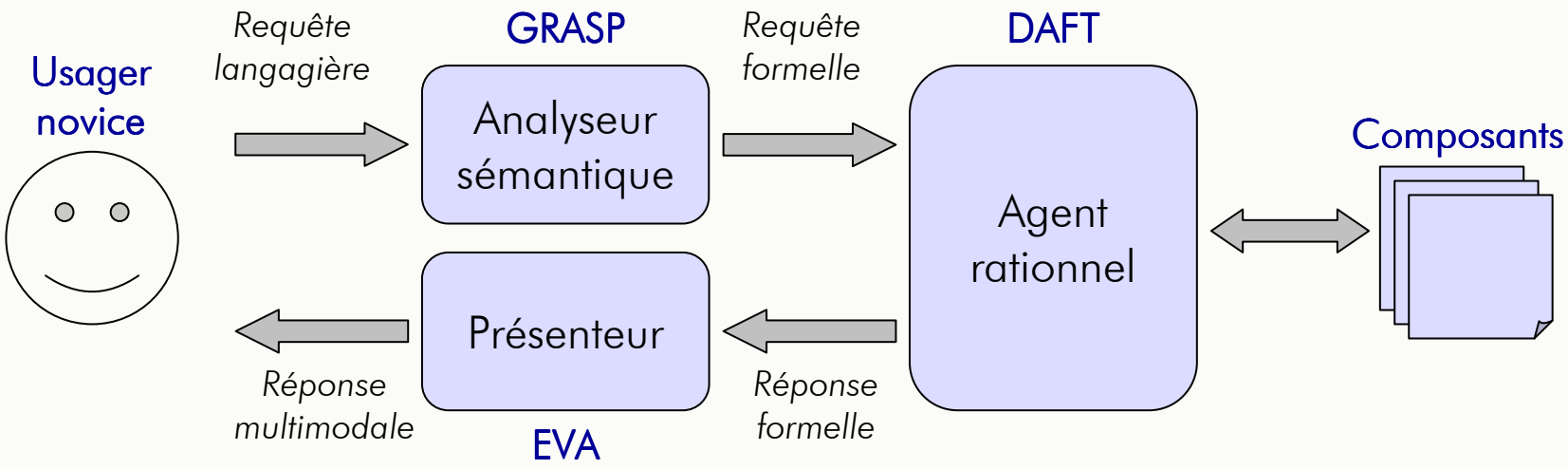
26 octobre 2006

WACA 2006

Contexte : le projet Daft

- Objectifs :
  - ACA dédiés à la fonction d'assistance :
    - Pour : les utilisateurs novices
    - Par : un raisonnement sur la structure et le fonctionnement des applications (aide contextuelle)

- Schéma d'architecture générale :



Problématique

- Approche corpus de la langue
  - Constitution d'un corpus
    - Pourquoi un corpus ?
    - Présentation du domaine de langue considéré
  - Couverture du corpus
    - Est-il représentatif du domaine étudié ?
    - Méthodes de construction
  - Spécificités du corpus
    - Est-il différent d'autres corpus disponibles ?
    - Études comparatives avec d'autres corpus
- Analyse du corpus
  - Catégorisation du corpus par ses activités
    - Est-il uniforme ?
    - Annotation manuelle de requêtes
  - Caractérisation des activités
    - Quelles sont leurs distinctions ?
    - Étude de différents critères de discrimination

Constitution du corpus Daft

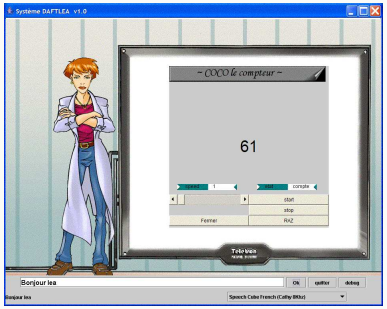
- On ne s'intéresse qu'à un domaine restreint de la langue : les Systèmes de Traitement de Requêtes d'Assistance, sur des requêtes isolées (pas de succession de dialogues).

- 8000 requêtes recueillies entre juin 2004 et septembre 2006.

- 2 méthodes complémentaires :
  - Des requêtes réelles d'utilisateurs (2/3) → empirisme
  - Des structures dialogiques génériques issues de thésaurus employées en contexte (1/3) → couverture

  qui permettent de garantir un corpus suffisant, c'est-à-dire représentatif du domaine étudié.

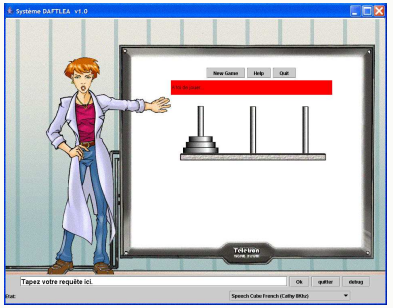
Les environnements de recueil du corpus



Application Java : compteur dont l'utilisateur contrôle la vitesse et le démarrage (thread)

Modèle : construit en parallèle de l'application

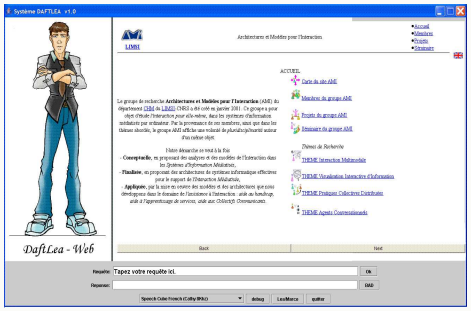
Public : restreint (~2/3 étudiants en informatique et ~1/3 d'usagers novices)



Application Java : jeu de tours de Hanoï (fonctionnement modal)

Modèle : construit a posteriori, par filtrage automatique

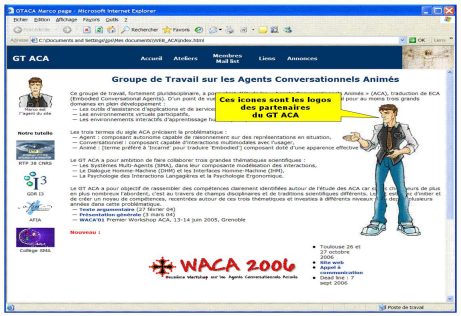
Public : restreint (~2/3 étudiants en informatique et ~1/3 d'usagers novices)



Site web : Version active du site du groupe AMI du LIMSI

Modèle : construit a posteriori, manuellement

Public : restreint (~2/3 étudiants en informatique et ~1/3 d'usagers novices)



Site web : Site internet du GT ACA

Modèle : construit a posteriori, manuellement

Public : Non contrôlé, ouvert à tous sur le web

Une vue du corpus Daft

…
a plus
a+
ah
Allez, bye
Allez ciao.
Alors ?
As-tu des amis?
auf viedersen
à l'aide !
bah!
barre toi de là
bidule
bon à rien !
bon week end
Bon.
Bonjour
Bonsoir
…

…
A quoi sers-tu ?
alors lâ t'es completement paumé !
appelle moi simplement Sylvie
as tu des informtion?
as tu entendu parler d'une expérimentation en cours ??
au sujet de cette page, que peut tu dire ?
avec ce corpus, tu sauras ce qu'est une anaphore …
à quoi penses tu?
be ouais tu comprends pas
ben alors reponds
bon j'en ai marre je me tire …
Bon je me casse. Bye.
bon y a rien â tirer de toi !!
bon, ça va, bonne année 2006
Bon, dis-moi plutôt ce que tu sais faire
bon, reviens à l apage d'accueil
bonjourmon vieux
…

Études comparatives de corpus

Daft : 8k requêtes d'assistance

Corpus de textes "généralistes"

Multitag [Paroubek, 2000] : phrases issues d'articles du journal Le Monde, de romans et d'essais.

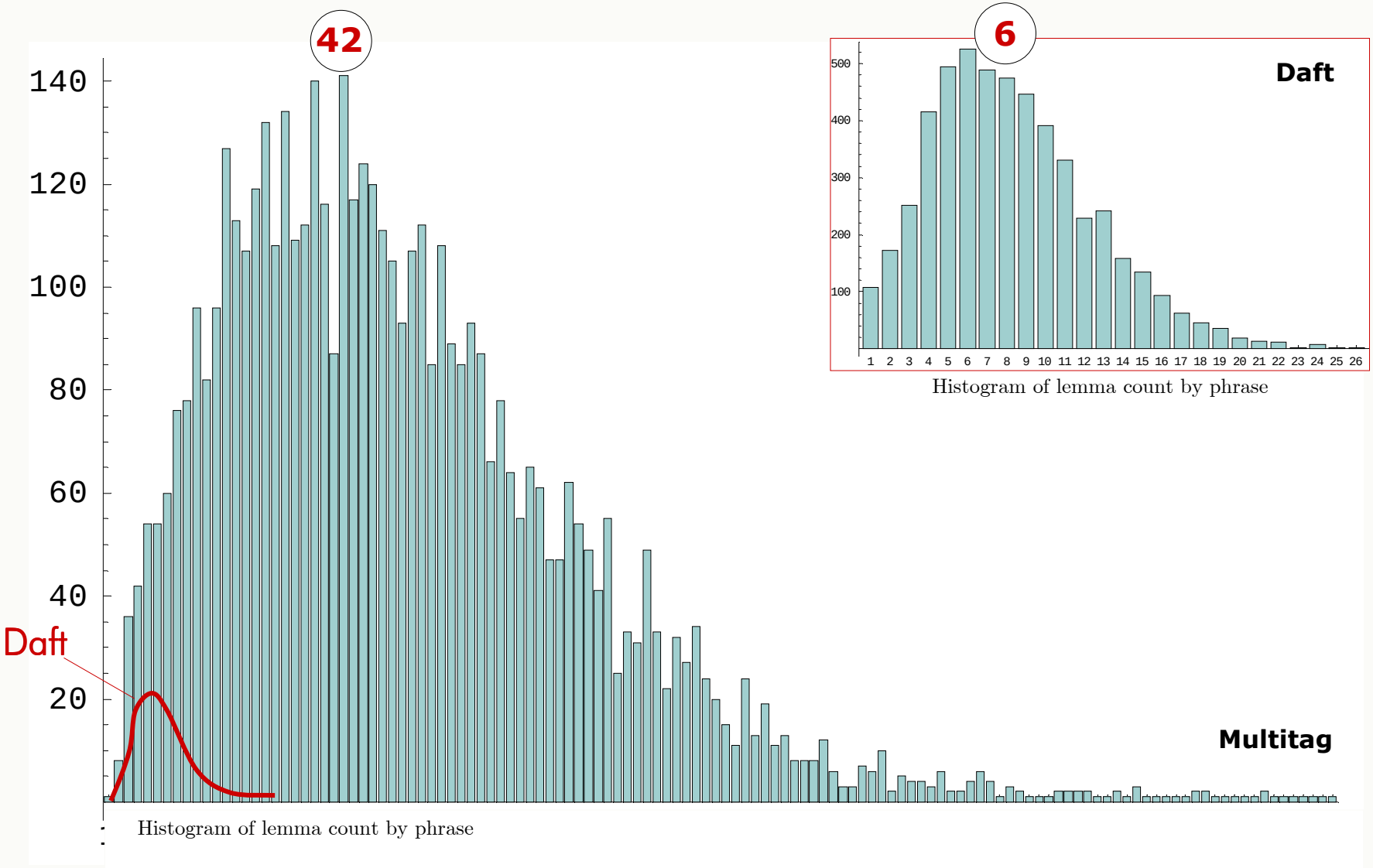
Corpus de dialogues humains "orientés tâche"

Switchboard [Jurafsky et al., 1998] : 200k conversations téléphoniques.

MapTask [Carletta et al., 1996] : 128 dialogues (reconstruction d'une carte).

Bugzilla [Ripoche, 2006]: 1,2M de commentaires de rapport de bugs issus de 128k rapports de défauts.

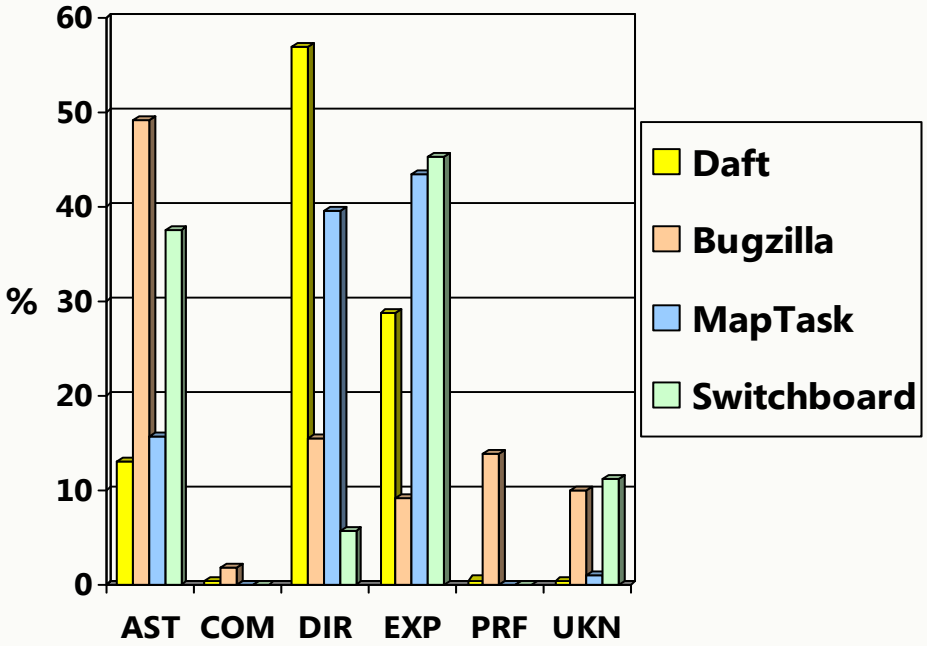
Comparaison Daft / Multitag



Comparaison de corpus orientés tâches

- Méthodologie de l'étude :
  - Justification : Méthode de comparaison de corpus annotés avec des taxonomies différentes.
  - Basée sur la notion de profil interactionnel [Ripoche, 2005]
- Définition :

  « *Représentation sous forme d'histogramme de la répartition des classes d'actes de dialogue dans un corpus donné.* »



- Résultats :
  - Une majorité de directifs (57%) : distinction entre DHM et DHH.
  - Plus d'expressifs que d'assertifs : l'utilisateur exprime plus ses sentiments que des faits « objectifs ».
  - Promissifs marginaux : soumission de l'agent à l'utilisateur, et non l'inverse.
- Conclusion : le corpus est spécifique, donc nécessaire.

Catégorisation du corpus Daft

- Méthodologie :

  Annotation manuelle des requêtes issues de deux sous-ensembles au 1/10e du corpus.

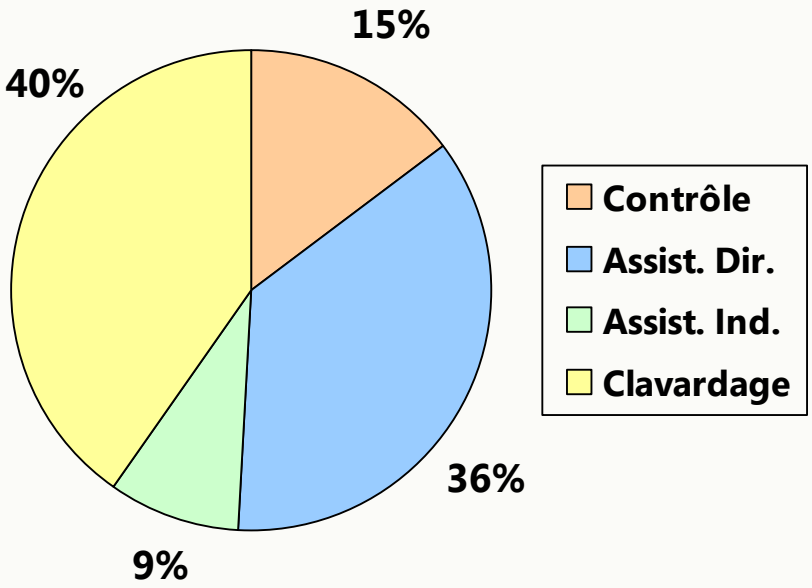
- Résultats :
  - Contrôle

    Commandes prédicatives via l'agent
  - Assistance directe

    Demandes d'aide explicites
  - Assistance indirecte

    Commentaires sous-entendant une demande d'aide (pragmatique)
  - Clavardage

    Interactions utilisateur-agent



- Conclusion :

  4 activités distinctes

  → 4 sous-corpus

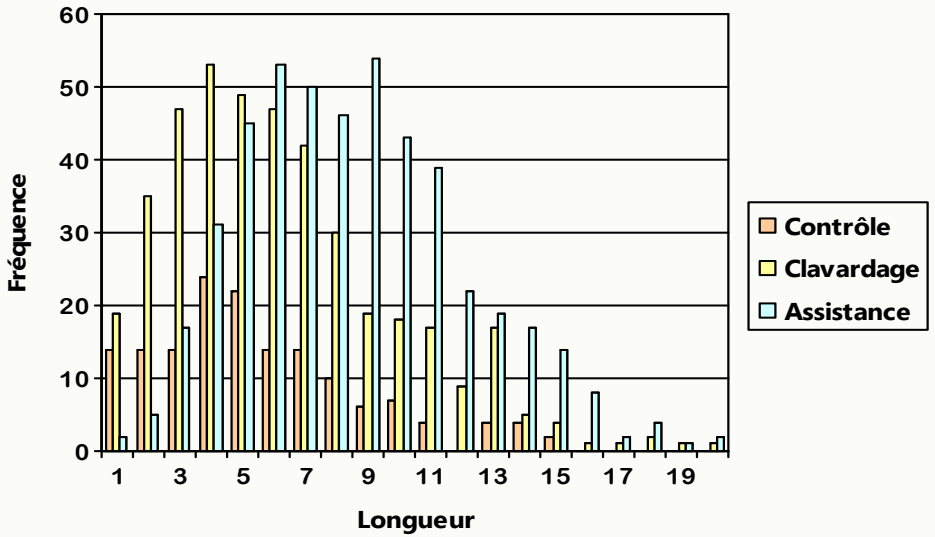
Caractérisation des activités

- Pourquoi les différencier ?
  - Pour détecter l'appartenance d'une requête à une des quatre activités.
  - Car les requêtes issues des quatre sous-corpus sont globalement de complexité croissante.

- Comment les différencier ?
  - Étude des distributions en fonction de la longueur des requêtes.
  - Étude des profils interactionnels des sous-corpus.
  - Étude de la sémantique des phrases retranscrites sous forme de requêtes formelles.

Caractérisation 1 : longueur des phrases

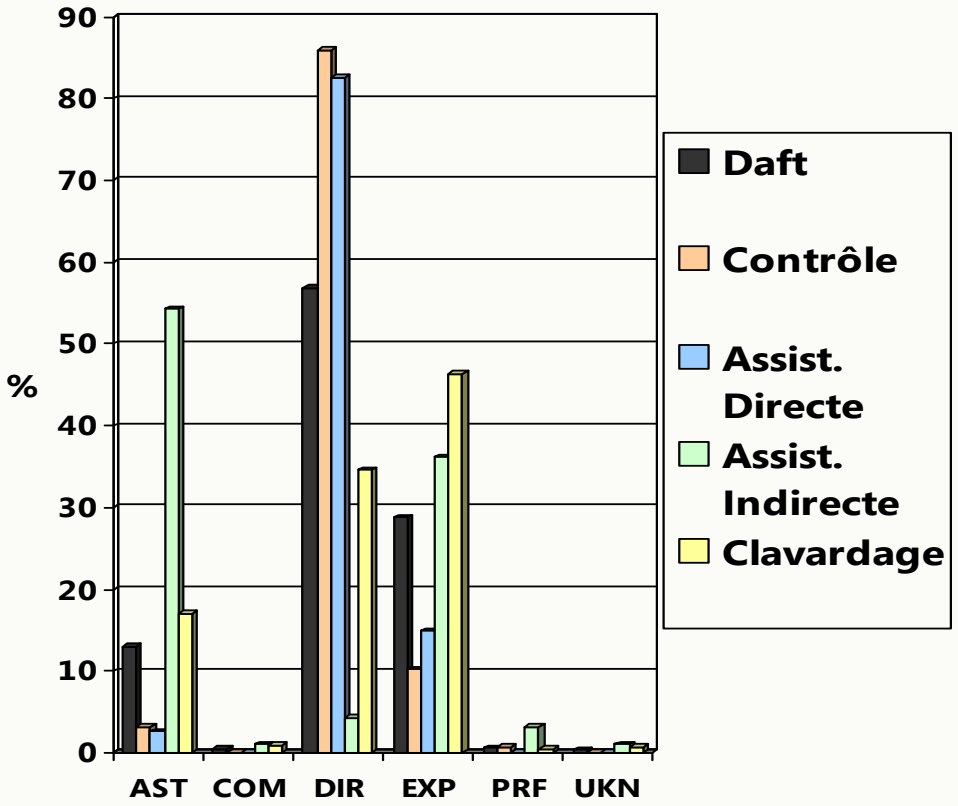
- Requêtes de contrôle sensiblement plus courtes.
- On peut approximer les distributions par des gaussiennes.
- Écart-types trop importants et moyennes trop proches pour classifier efficacement.



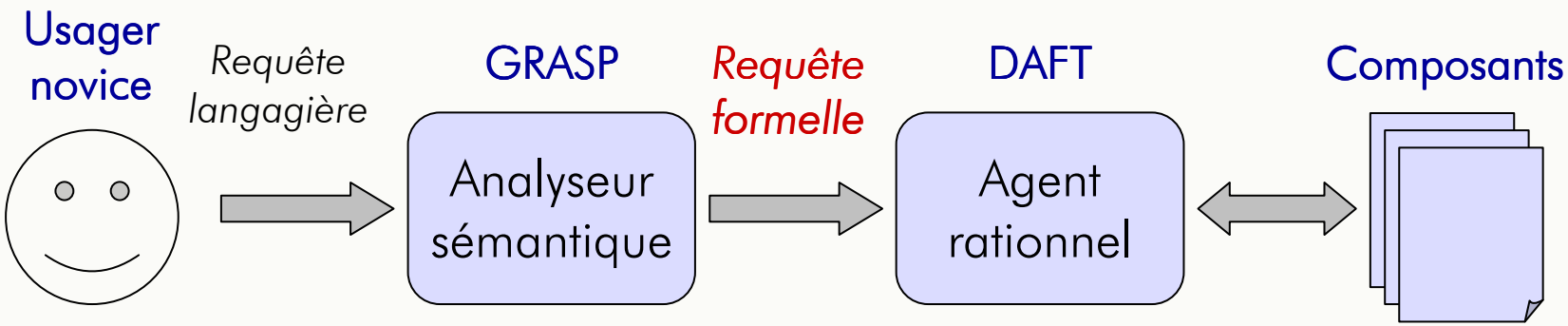
Sous-corpus	Ctrl	Ass. Dir.	Ass. Ind.	Clav.
Moyenne	5,44	8,01	9,90	6,01
Écart-type	3,36	3,54	3,30	3,62

Caractérisation 2 : profils interactionnels

- Bonne distinction entre l'assistance directe et indirecte.
- Peu de différence entre le contrôle et l'assistance directe.
- Méthode inefficace seule.
- Détermination automatique non triviale.



Caractérisation 3 : requêtes formelles



- Le langage formel DAFT sert :
  - Pour la représentation de la sémantique des requêtes des usagers.
  - De support aux raisonnements de l'agent rationnel.

Caractérisation 3 : requêtes formelles

- Les requêtes sont de la forme :

$$M_1(\ldots M_n(\ c_1 = P_1(c'_1 = R_1, \ldots, c'_l = R_l), \ldots, c_m = P_m(\ldots)) \ldots)$$

  - **Modalités** (M_i) ≈ actes de dialogue (« obligation », « avoir peur de », etc.)
  - **Prédicats** (P_i) = verbes d'action ou de prédication (« modifier », « être prêt », etc.)
  - **Références** (R_i) ≈ GN (« le petit bouton », etc.) – Réf. Extensionnelles Associatives

- Exemples de requêtes :

Glisser le disque de droite à gauche	Bouger(objet="le disque",origine="droite", destination="gauche")
d'après toi, y a t-il des fonctions d'annulation dans cette application ?	ASK(KNOWLEDGE(of=s, about=EXISTENCE(time=2, of=FUNCTION(doing=Annuler(), in="cette appli...")))
j'ai peur qu'il n'y ait pas moyen de changer la taille de la police qui est bien trop petite	FEAR(agent=u, fear=NEG(POSSIBILITY(todo=Modifier(objet="la police...trop petite",propriété="taille"))))

Caractérisation 3 : requêtes formelles

Sous-corpus	Nombre moyen de prédicats	Nombre moyen de modalités
Contrôle	0,96	0,25
Assistance Directe	0,54	2,40
Assistance Indirecte	0,59	2,22
Clavardage	0,19	0,85

- Le nombre de prédicats et modalités contenus dans une requête varie en fonction de son sous-corpus.
- Cette méthode permet de distinguer clairement les requêtes de contrôle, d'assistance et de clavardage.
- Elle ne discrimine pas l'assistance directe de l'assistance indirecte.

Conclusion

- Le corpus Daft :
  - Se distingue des autres corpus de nature similaire disponibles.
  - Offre une bonne couverture du domaine, en dépit de sa taille restreinte, grâce à une constitution mixte.
- On peut distinguer 4 activités distinctes :
  - Contrôle de l'application
  - Demande d'assistance directe
  - Demande d'assistance indirecte
  - Clavardage avec l'agent (dû à la personification)
- Ces activités sont :
  - Facilement distinguables par un annotateur humain,
  - Difficiles à classifier automatiquement.

  On pourra envisager de le faire en combinant analyse des requêtes formelles et profils interactionnels.