Vers la construction d'ontologies dynamiques par un système multi-agent

Kévin Ottens, Valérie Camps & Pierre Glize

IRIT - Université Paul Sabatier

JFSMA 2007 - 17 octobre - Carcassonne

Plan

Plan

Motivations

Web sémantique

- Besoin de connaissances structurées
- Effort de production prohibitif

Construction d'ontologies à partir de textes

- Réduit l'effort de production
- Difficile à maintenir et réviser à cause de la masse d'information
- *Ontologies dynamiques* ou pointer ce qui est pertinent et nouveau dans les textes

Motivations

Web sémantique

- Besoin de connaissances structurées
- Effort de production prohibitif

Construction d'ontologies à partir de textes

- Réduit l'effort de production
- Difficile à maintenir et réviser à cause de la masse d'information
- *Ontologies dynamiques* ou pointer ce qui est pertinent et nouveau dans les textes

Buts de ces travaux

Semi-automatiser la construction

- Traitement automatique des langues
- Interaction avec l'ontologue

Dynamic **O**ntologies

- Structure elle-même dynamique
- Conception vivante
- Système multi-agent
  - Chaque agent se place dans l'organisation
  - Chaque agent est porteur du processus de construction
  - Le système est l'ontologie

Buts de ces travaux

Semi-automatiser la construction

- Traitement automatique des langues
- Interaction avec l'ontologue

Dynamo

- Structure elle-même dynamique
- Conception vivante
- Système multi-agent
  - Chaque agent se place dans l'organisation
  - Chaque agent est porteur du processus de construction
  - Le système est l'ontologie

Buts de ces travaux

Semi-automatiser la construction

- Traitement automatique des langues
- Interaction avec l'ontologue

Dynamo

- Structure elle-même dynamique
- Conception vivante
- Système multi-agent
  - Chaque agent se place dans l'organisation
  - Chaque agent est porteur du processus de construction
  - Le système est l'ontologie

Buts de ces travaux

Semi-automatiser la construction

- Traitement automatique des langues
- Interaction avec l'ontologue

Dynamo

- Structure elle-même dynamique
- Conception vivante
- Système multi-agent
  - Chaque agent se place dans l'organisation
  - Chaque agent est porteur du processus de construction
  - Le système est l'ontologie

Plan

Conception : « Le SMA est l'ontologie »

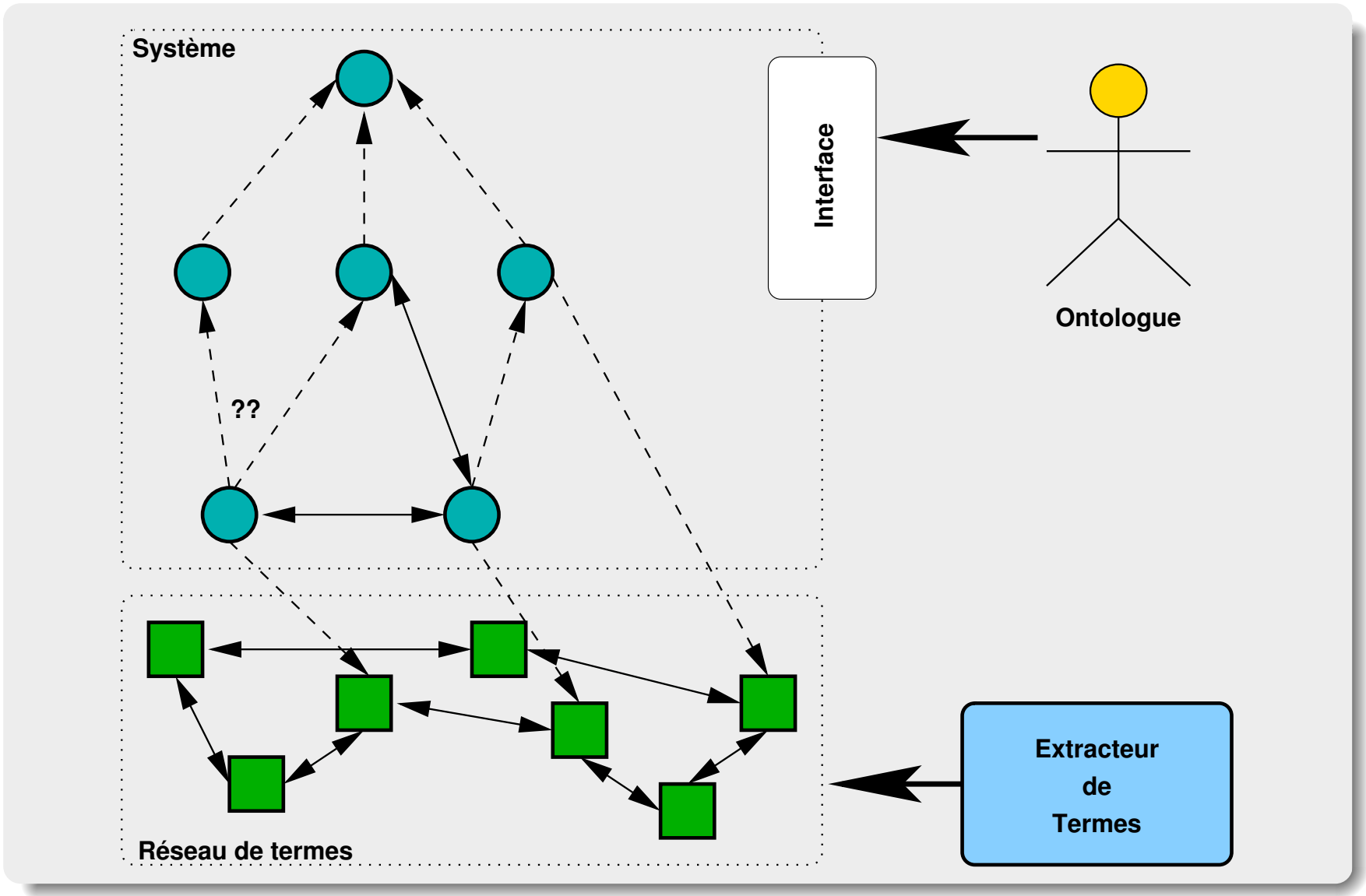
Vers moins d'intervention manuelle

- Exploiter les informations en provenance de l'extracteur de termes (Syntex)
- Faire valider une organisation conceptuelle
- Réagir aux modifications de l'ontologue

Un autre point de vue sur l'ontologie

- Ontologie = équilibre entre les agents-concepts qui la composent
- Modification de l'ontologie = perturbation de l'équilibre
- Ontologie dynamique = processus auto-organisé cherchant à résorber les perturbations

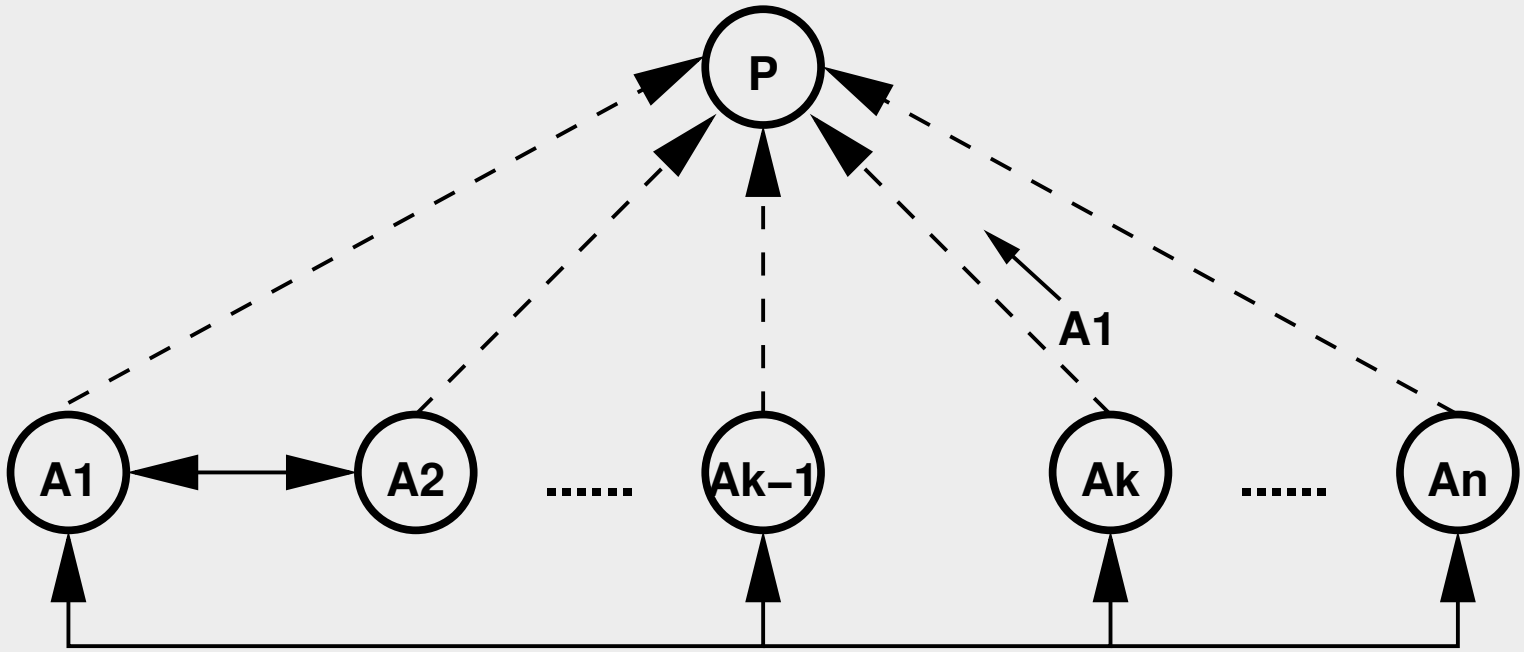
Architecture



Plan

Algorithme distribué de classification

Vue locale

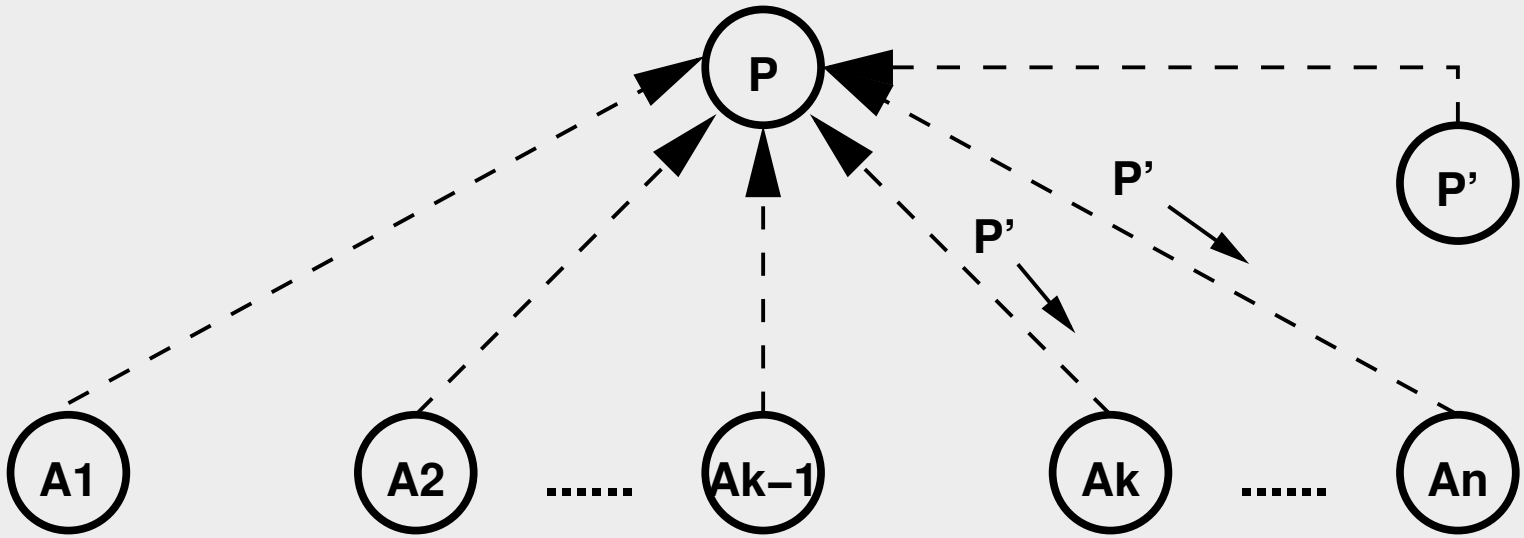


Etapes

1. Évaluation des similarités et « votes »
2. Partitionnement et création du niveau intermédiaire
3. Changement de parent

Algorithme distribué de classification

Vue locale

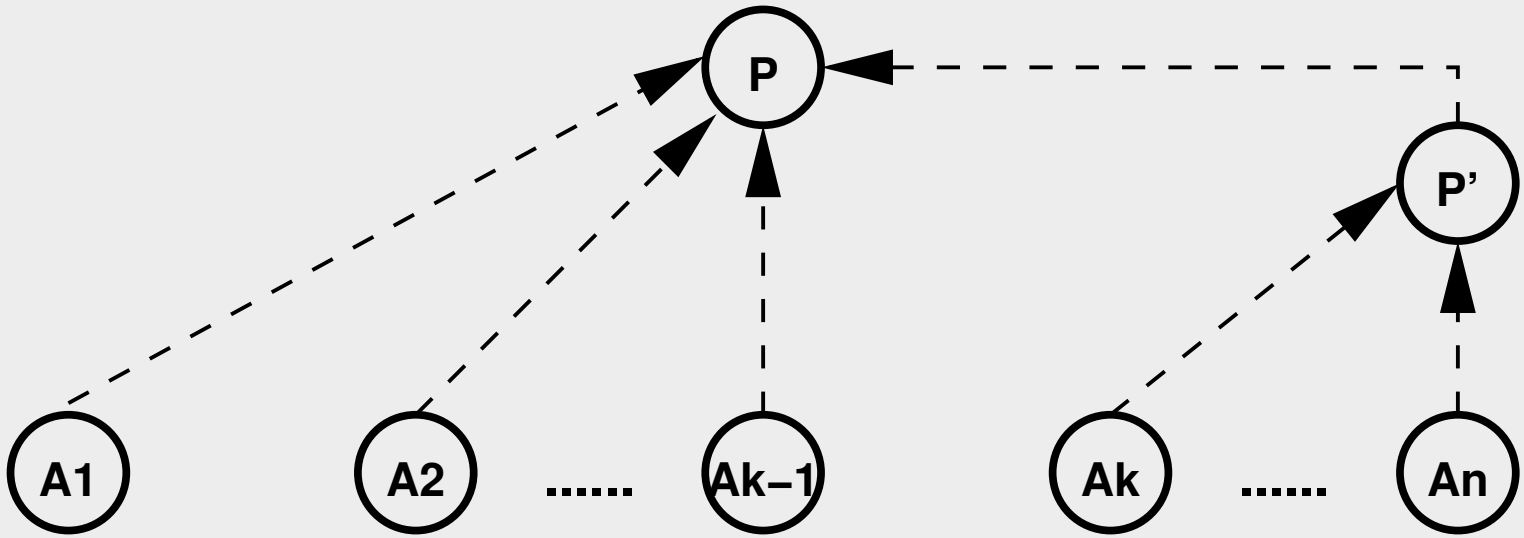


Etapes

1. Évaluation des similarités et « votes »
2. Partitionnement et création du niveau intermédiaire
3. Changement de parent

Algorithme distribué de classification

Vue locale

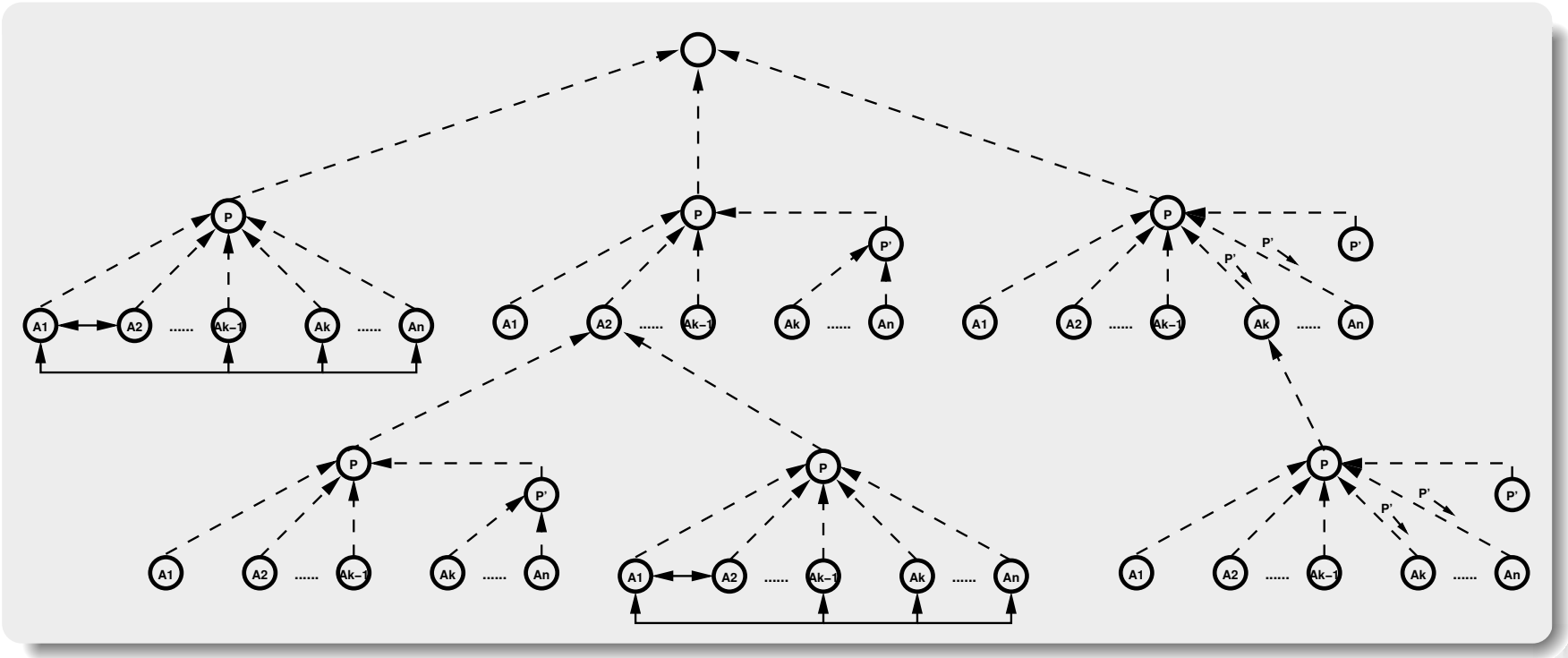


Etapes

1. Évaluation des similarités et « votes »
2. Partitionnement et création du niveau intermédiaire
3. Changement de parent

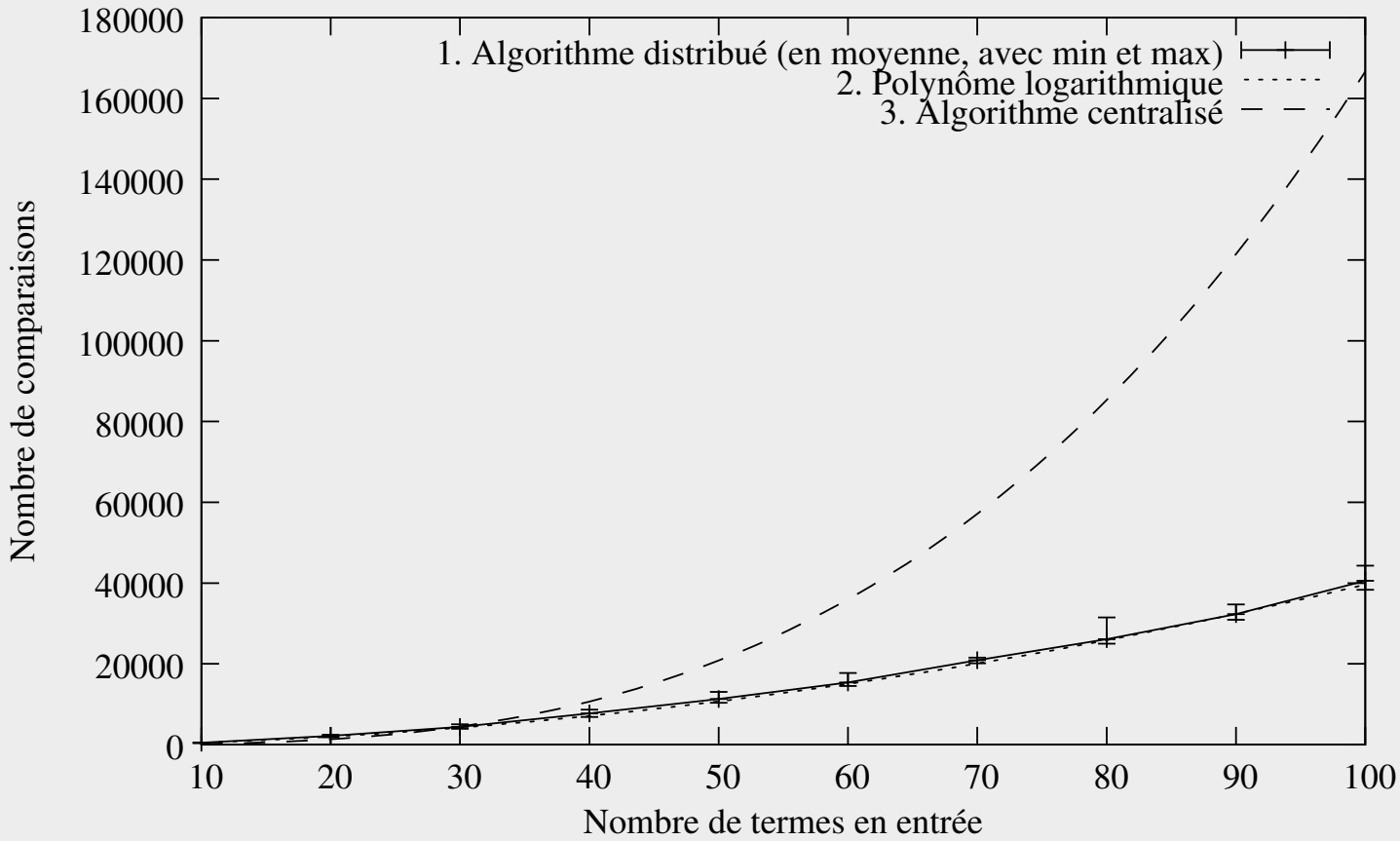
Algorithme distribué de classification

Vue globale



Algorithme distribué de classification

Complexité expérimentale



- Complexité en moyenne: $O(n^2 log(n))$
- Écart-type maximum: environ 5%

Algorithme distribué de classification

Point de vue qualitatif

Exécution automatique

- Vue permanente de la hiérarchie en construction
- Permet d'obtenir un « premier brouillon »

Intervention de l'ontologue

- Pas d'ajustement de l'algorithme requis
- Dynamicité, la structure est révisée

Règle de couverture en tête

Comportement recherché

Observations

- Il existe des paires de termes où la similitude n'est pas définie
- Les ontologues ont des heuristiques spécifiques à la structuration des feuilles

Buts

- Pouvoir placer ces termes
- Implémenter une heuristique similaire

Fonction d'adéquation du parent

- Le meilleur parent pour E est l'agent P qui maximise $a(P, E)$
- *Quand un agent E est insatisfait de son père P, il évalue $a(F_i, E)$ avec tous ses frères (notés F_i). Celui maximisant $a(F_i, E)$ est alors choisi comme nouveau parent*

Gérer plusieurs critères

Guide

Comment?

- Rester simple
  - Critères locaux
  - Valeurs nominales pour ces critères
- AMAS : utiliser l'heuristique de la coopération

Coopération

- Minimiser la non-coopération
- Système à priorités
  - Déterminer tous les problèmes en cours
  - Trouver le plus critique
  - Essayer de le résoudre

Gérer plusieurs critères

Implémentation

Minimiser la non-coopération

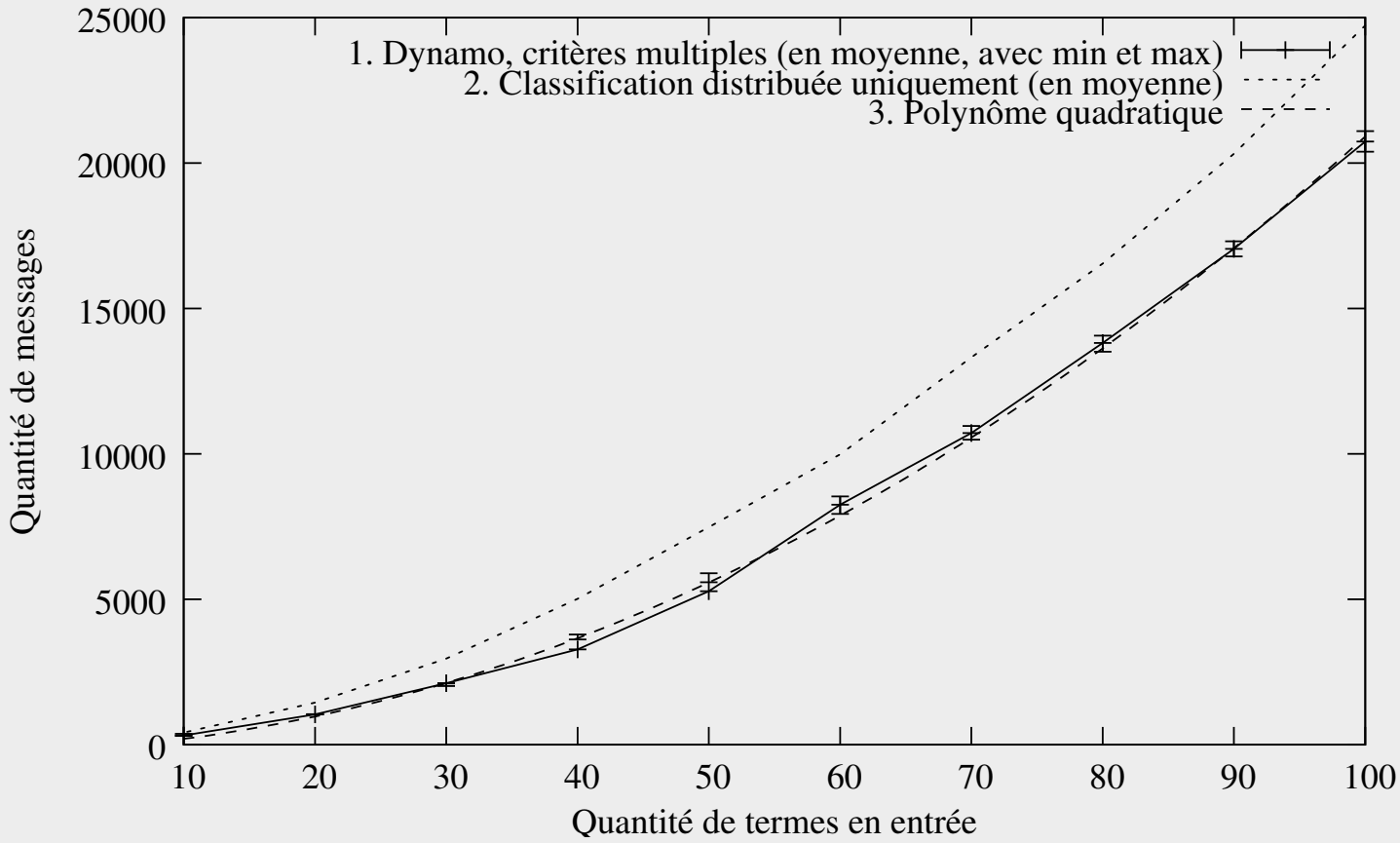
- $\mu_T(A)$: niveau de non-coopération de « couverture en tête »pour A
- $\mu_F(A)$: niveau de non-coopération de la « fratrie »pour A
- $\mu_M(A)$: niveau de non-coopération par « message »pour A
- $\mu(A) = max(\mu_T(A), \mu_F(A), \mu_M(A))$

Prendre en charge le problème le plus critique

- $\mu(A) = \mu_T(A) \rightarrow$ Essayer de trouver un meilleur parent
- $\mu(A) = \mu_F(A) \rightarrow$ Améliorer la structure par la classification
- $\mu(A) = \mu_M(A) \rightarrow$ Traiter le message

Gérer plusieurs critères

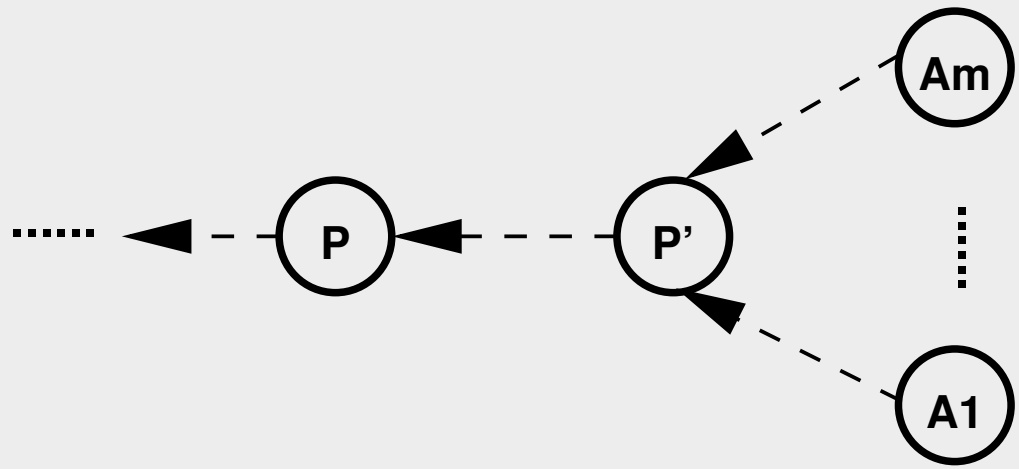
Impact sur la complexité



- Complexité en moyenne: $O(n^2)$
- Écart-type maximum: environ $0,6\%$

Plan

Nettoyage de l'arborescence

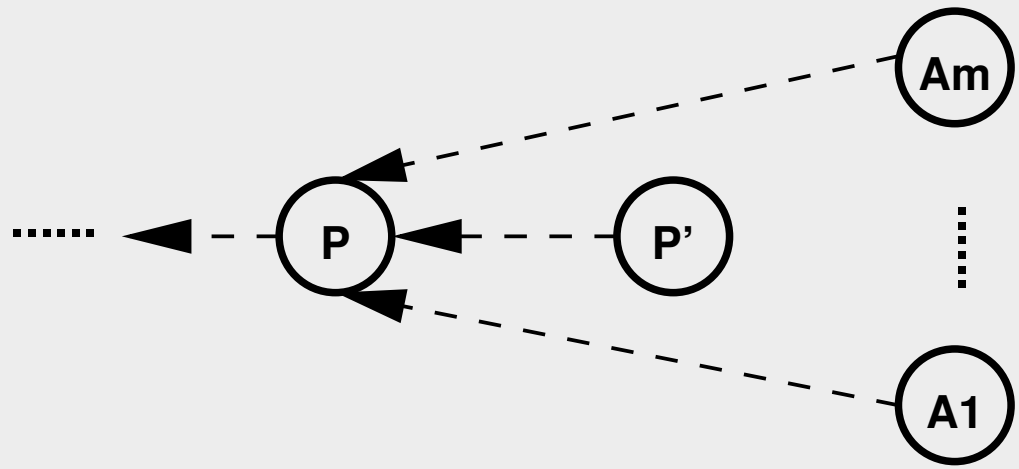


Etapes

1. P' n'a aucun frère, il propose à ses fils de remonter
2. P' n'a aucun fils, et n'est représenté par aucun terme, il doit disparaître

- Complexité : $O(1)$

Nettoyage de l'arborescence

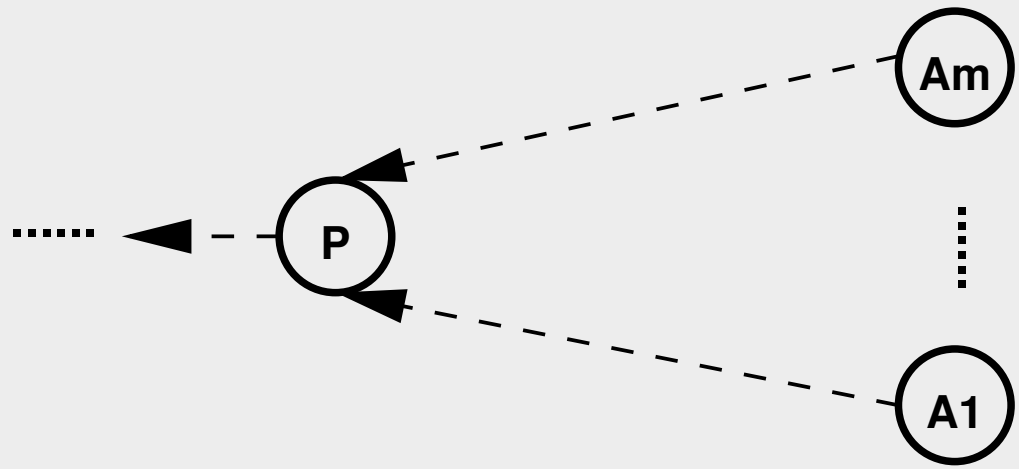


Etapes

1. P' n'a aucun frère, il propose à ses fils de remonter
2. P' n'a aucun fils, et n'est représenté par aucun terme, il doit disparaître

- Complexité : $O(1)$

Nettoyage de l'arborescence

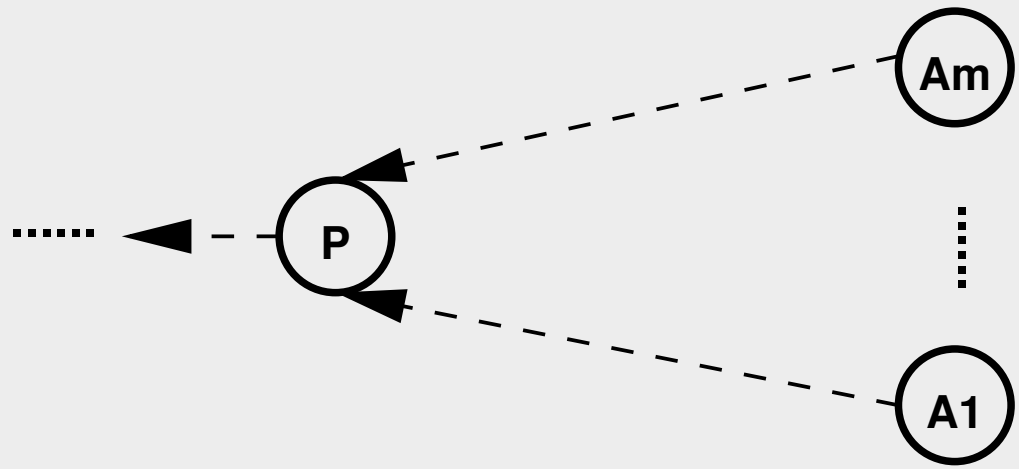


Etapes

1. P' n'a aucun frère, il propose à ses fils de remonter
2. P' n'a aucun fils, et n'est représenté par aucun terme, il doit disparaître

- Complexité : $O(1)$

Nettoyage de l'arborescence



Etapes

1. P' n'a aucun frère, il propose à ses fils de remonter
2. P' n'a aucun fils, et n'est représenté par aucun terme, il doit disparaître

- Complexité : $O(1)$

Passage aux arbres n-aires

Comportement recherché

Observations

- La classification pousse à un arbre binaire
- Le profil de l'arbre est conditionné par la stratégie de vote des agents

But

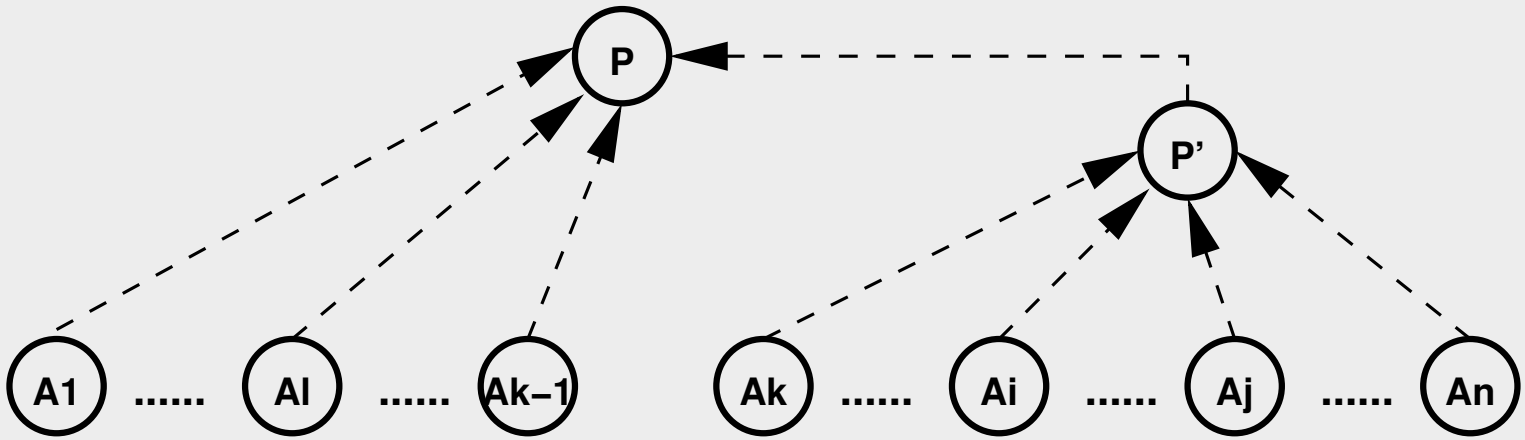
- Permettre d'obtenir des arbres n-aire

Nouvelle stratégie de vote

- Introduction d'une tolérance ε sur la similitude
- Préférence partielle lors des votes en fonction de ε

Passage aux arbres n-aires

Détails

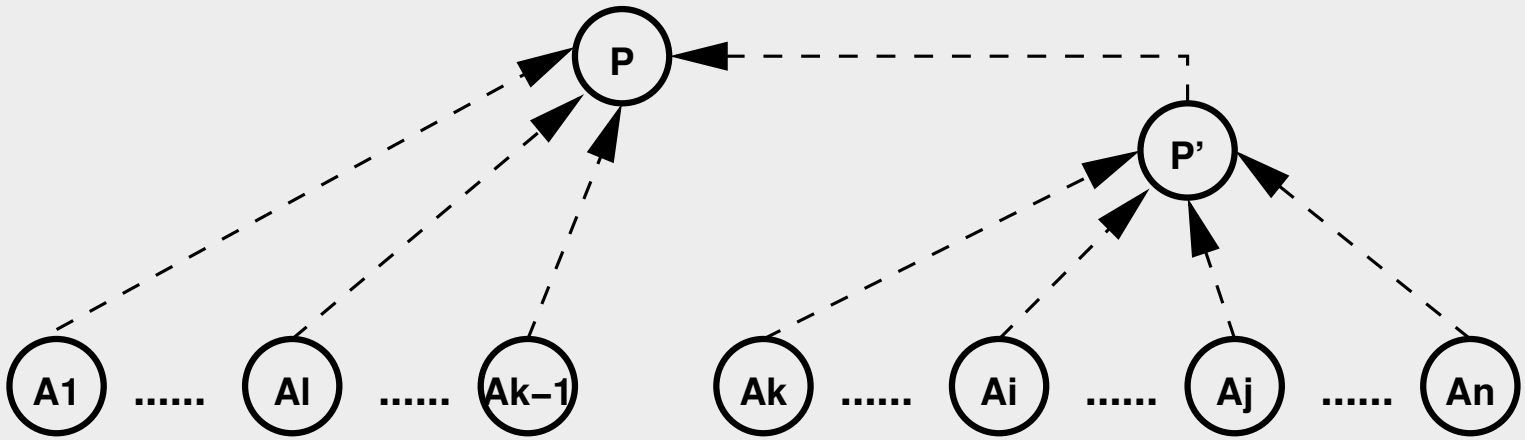


Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l)$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils $\rightarrow$ baisser $\varepsilon_{P'}$
  - Trop peu de fils $\rightarrow$ augmenter $\varepsilon_{P'}$

Passage aux arbres n-aires

Détails

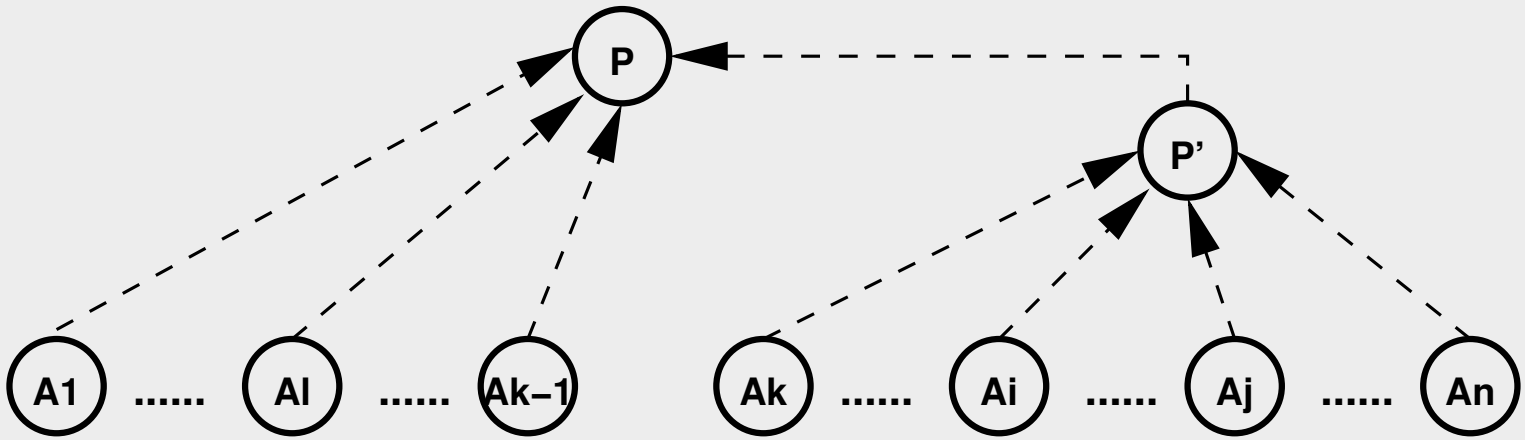


Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l) + \varepsilon$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils $\rightarrow$ baisser $\varepsilon_{P'}$
  - Trop peu de fils $\rightarrow$ augmenter $\varepsilon_{P'}$

Passage aux arbres n-aires

Détails

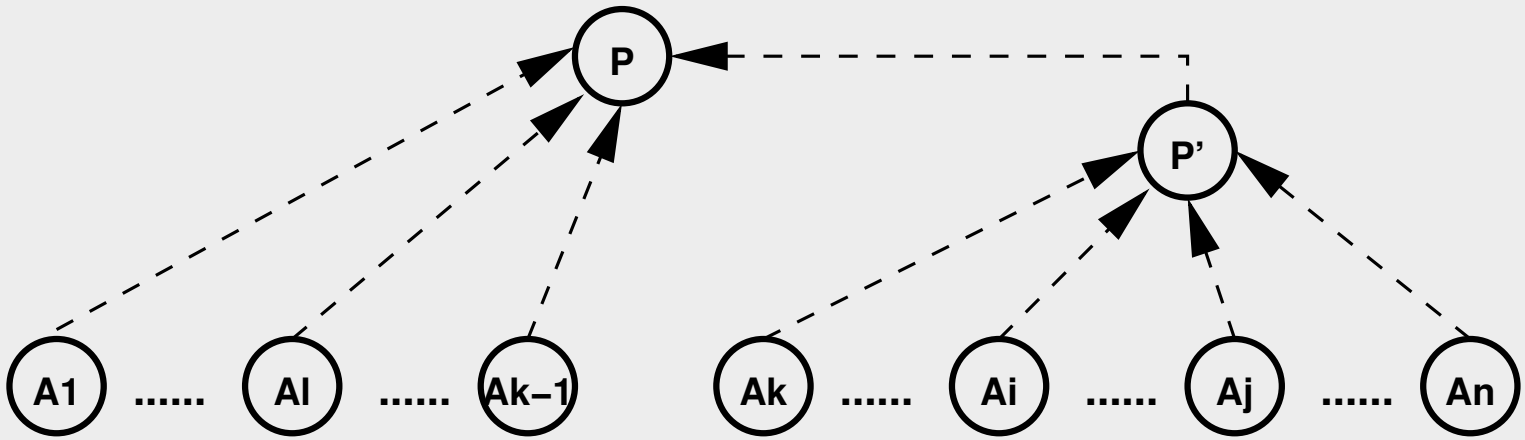


Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l) + \varepsilon_{P'}$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils → baisser $\varepsilon_{P'}$
  - Trop peu de fils → augmenter $\varepsilon_{P'}$

Passage aux arbres n-aires

Détails

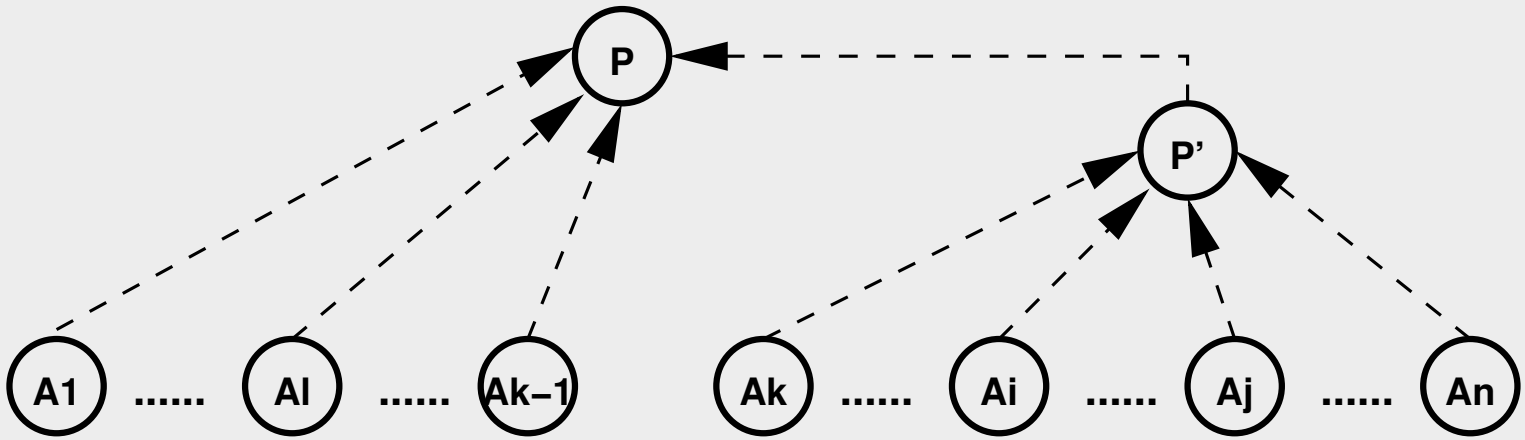


Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l) + \varepsilon_{P'}$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils $\rightarrow$ baisser $\varepsilon_{P'}$
  - Trop peu de fils $\rightarrow$ augmenter $\varepsilon_{P'}$

Passage aux arbres n-aires

Détails

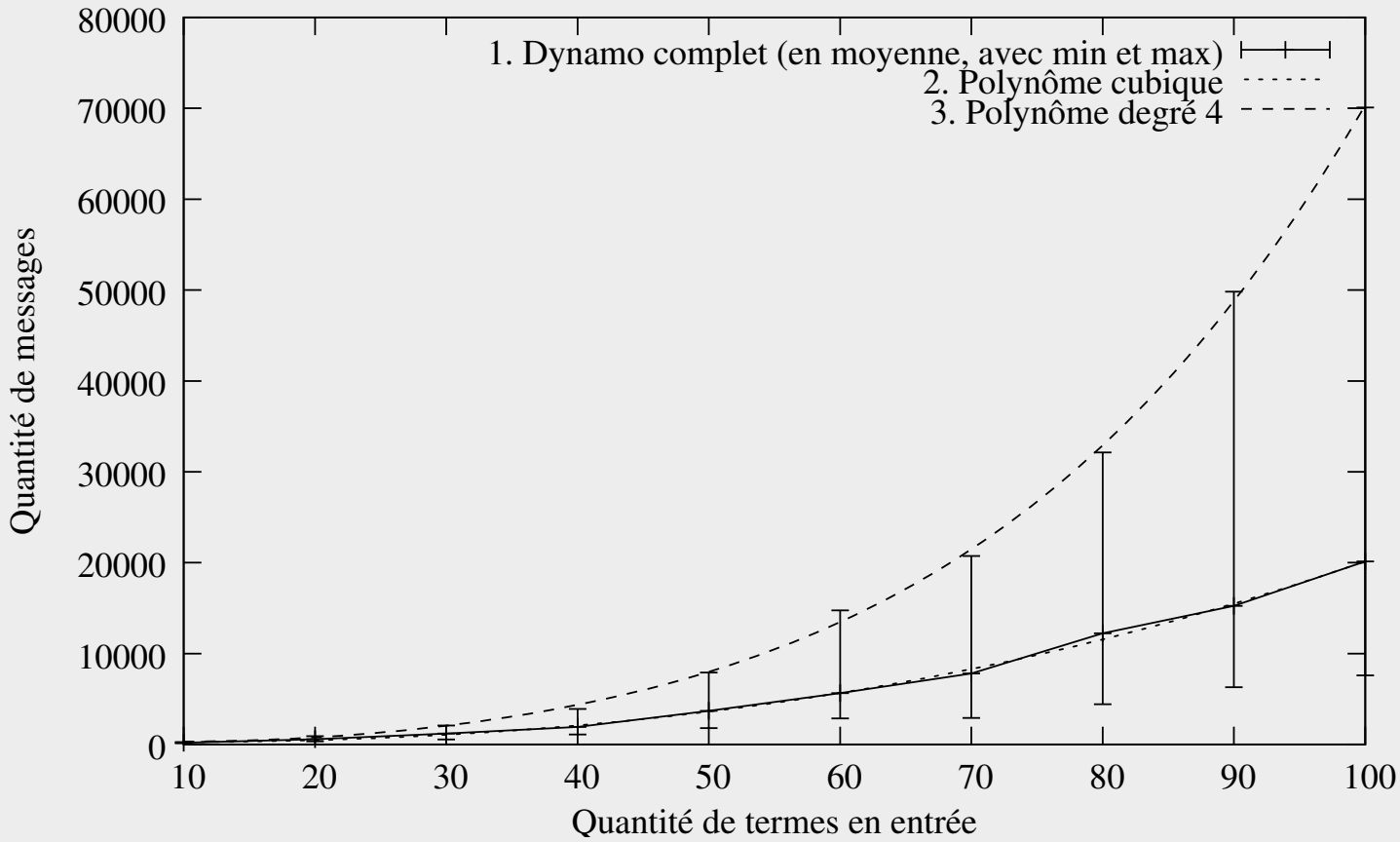


Propriété sur la similitude

- $sim(A_i, A_j) > sim(A_i, A_l) + \varepsilon_{P'}$
- Introduction d'une tolérance
- Tolérance par niveau
- Prise en compte du facteur de branchement
  - Trop de fils $\rightarrow$ baisser $\varepsilon_{P'}$
  - Trop peu de fils $\rightarrow$ augmenter $\varepsilon_{P'}$

Passage aux arbres n-aires

Impact sur la complexité



- Complexité en moyenne: $O(n^3)$
- Complexité au pire cas: $O(n^4)$

Plan

Discussion

Avantages de l'approche

- Couplage système/ontologue plus aisé
- Distribution possible sur un réseau

Limites actuelles

- Résultats dépendants de l'ordre d'ajout
- Manque de critères pour guider l'ontologue et le système

Perspectives

Ingénierie des connaissances

- Dégager les relations transverses de l'ontologie
- Faciliter la réutilisation
- Plus de travaux de validation et d'ergonomie

Systèmes multi-agents

- Améliorer l'algorithme de classification
  - Supprimer la dépendance sur l'ordre d'ajout
  - Optimiser
- Utiliser l'algorithme dans d'autres domaines

Travaux en cours, Conclusion

En cours...

- Evaluation du système sur d'autres corpus
- Evaluation de l'apport à une application de recherche d'information

Conclusion

- Une structure dynamique est possible dans ce domaine
- Les performances sont acceptables
- Des efforts sont encore nécessaires...

Questions ?

Kévin Ottens

ottens@irit.fr
(Plus pour longtemps)