Interfaces adaptatives : apprendre pour prédire le comportement de l'utilisateur

Jean-David Ruvini
e-lab Bouygues
jdruvini@bouygues.com

Plan

S'adapter?

Par apprentissage?

Tour d'horizon : modélisation, performances

- Prédire des commandes textuelles
- Compléter des formulaires
- Filtrer des emails ou des news
- Automatiser des séquences d'actions
- Recommander des liens
- Recommander des produits

Une approche viable?

Résumé

Perspectives

S’adapter

S’adapter?

Par apprentissage?

Tour d’horizon : modélisation, performances

- Prédire des commandes textuelles
- Compléter des formulaires
- Filtrer des emails ou des news
- Automatiser des séquences d’actions
- Recommander des liens
- Recommander des produits

Une approche viable?

Résumé

Perspectives

S'adapter?

Interfaces adaptatives (Dieterich et al 93)

- Objectifs :
  - Présenter des info pertinentes
  - Faciliter / accélérer l'utilisation
  - Satisfaire des groupes hétérogènes d'utilisateurs
  - S'adapter aux compétences des utilisateurs
- Mécanismes :
  - Suggérer des commandes (erreurs ou plans sub-optimaux)
  - Basculer entre différents types d'interaction
  - Modifier l'organisation, la présentation
  - Créer des macro-commandes
  - …

S'adapter?

- Mode :
  - utilisateur passif / actif
- Stratégie :
  - Avant la première session / Au cours des sessions / A la fin des sessions
- Composants :
  - Modèles des tâches / modèles du dialogue / modèles de l'utilisateur
- Cible :
  - Utilisateur typique / groupe d'utilisateurs / utilisateur individuel (habitudes, préférences, connaissances, expérience…)
- Niveau :
  - Invisible (logique) / visible (physique)

Par apprentissage?

S'adapter?

Par apprentissage?

Tour d'horizon : modélisation, performances

- Prédire des commandes textuelles
- Compléter des formulaires
- Filtrer des emails ou des news
- Automatiser des séquences d'actions
- Recommander des liens
- Recommander des produits

Une approche viable?

Résumé

Perspectives

Par apprentissage?

Les utilisateurs ont tendance à se répéter

Les utilisateurs ont tendance à se répéter

Construire un modèle prédictif des utilisateurs
en généralisant les corrélations entre

- Les situations que les utilisateurs ont rencontrées dans le passé
- Les choix qu'ils ont fait

Tour d'horizon

S'adapter?

Par apprentissage?

Tour d'horizon : modélisation, performances

- Prédire des commandes textuelles
- Compléter des formulaires
- Filtrer des emails ou des news
- Automatiser des séquences d'actions
- Recommander des liens
- Recommander des produits

Une approche viable?

Résumé

Perspectives

Tour d'horizon

Prédire des commandes

Travaux pionniers (Crochetiere et al 74):

- Outils d'aide à la communication pour les personnes souffrant de handicaps physiques
- Prédiction de lettres

  P(lettre|2 lettres) ≈ 75% (Shannon 51)

Unix

- The Reactive Keyboard (Darragh et Witten 91)
- redondant à 70% (Greenberg et al 93)

IPAM

(Davison et Hirsh 97, 98)

Prédiction de commandes Unix

Etant donné une commande, prédit la suivante

- Enregistre les commandes saisies
- Pour chaque paire (C,C') de commandes, maintient P(C'|C) (Markov ordre 1)
- Distribution par défaut pour les nouvelles commandes

Prédiction sur la ligne de commande

IPAM : performances

Evaluation: 168 utilisat., 287 000 commandes

- 40% sur une suggestion
- 63% sur 3 suggestions

- 38% pour C4.5 et NBC

IPAM : peut-on faire mieux?

Korvemaker et Greiner (2000)

– Markov ordre 2, généralisation des paramètres, utilisation code erreur… pas mieux!

Gorniak et Poole (2000)

– Considère la longueur moyenne des séquences
– Légèrement meilleur que IPAM !
– Suggère que Markov ne tient pas

ClipBoard
(Motoda 97)

Prédiction de commandes Unix

- Historique des commandes représenté sous forme d'un graphe orienté qui modélise
  - L'enchaînement des commandes
  - Le flot des données entre les commandes
- Graph based induction (Yoshida et Motada 95) et arbres de décisions
- 57% sur commandes complètes

Prédire des commandes : résumé

Modèle prédictif spécifique à un utilisateur

Algo basiques ≈ algo plus sophistiqués ?

Tour d'horizon

Compléter des formulaires

Formulaires administratifs
(Hermens et Schlimmer 93)

Le système propose des valeurs par défaut

- Prédire la valeur d'un champ à partir des champs précédents

- Exemples : formulaires complétés
- ID4

- 80% de prédictions correctes
- Gain de 87% en terme de touches pressées

CAP
(Dent et Mitchell et al 92, 94)

Aide à la saisie de rendez-vous dans un agenda

- Prédire heure, durée, lieu et jour en fonction d'une description du rendez-vous

- Exemples : les 180 rendez vous les plus récents
- ID3, chaque nuit

- 45% sur heure, lieu et jour, 30% sur durée
- 90% en limitant le nombre de prédictions à 30%

Compléter des formulaires : résumé

Performances dépendent de l'application

Assistance partielle

Tour d’horizon

Filtrer des emails ou des news

SwiftFile
(Segal et Kephart 99,00)

Aide au classement de mails dans LotusNotes

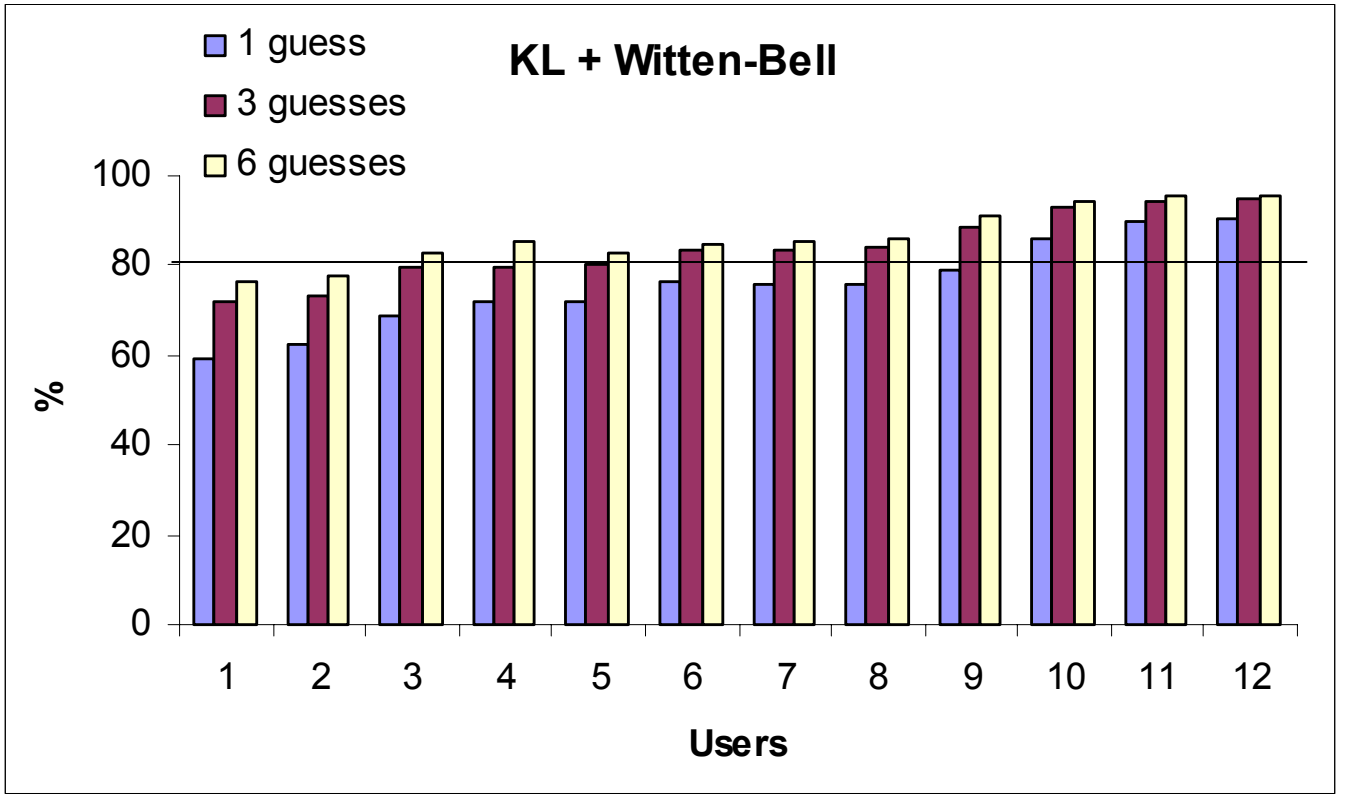
- Calcule P(dossier | message)
- Variante incrémentale de TFIDF

- Système passif, se contente de suggérer 3 dossiers
  - Augmente productivité
  - Réduit risque d'erreur

Smartlook
(Ruvini et Gabriel 01)

Démo

Smartlook : performances
(Ruvini et Gabriel 01)



Incrémentalité : +5% (Segal et Kephart 00)

Autres résultats
(Brutlag et Meek 01)

Comparaison de SVM, TFIDF et Unigram

- Taux de prédiction fonction monotone de la taille des dossiers
- TFIDF meilleur sur petits dossiers
- Unigram et SVM meilleurs sur gros dossiers

- Plus de différences entre boîtes aux lettres qu'entre méthodes

- Le corps des messages peut être ignoré ?

News Dude
(Billsus et Pazzani 99)

Lire des news intéressantes à l'utilisateur

- NewsDude propose des news à l'utilisateur
- L'utilisateur note les news proposées (1 … 4)

- Apprentissage quotidien sur Palm VII
- NBC pour long terme, k-NN pour court terme

- Performances : 70% après 3 jours (10 utilisateurs, 3000 news)

- NewsWeeder (Lang 95) : 50%

Notification de mail
(Horvitz et al. 99)

Notification de messages importants

- Compare le coût estimé de l'interruption à celui du report de la notification
- Exemples fournis par l'utilisateur
- Bayes net
  - en voyage, en réunion au bureau, contenu (Free !, Only $, J'ai besoin de…), heure, proximité de deadlines, rendez-vous prévus, etc.
- Mackassy et Hirsh et al. (99)
  - NBC, TFIDF, Ripper et Winnow équivalents
  - Break-even point 53%

Filtrer des emails ou des news : résumé

Présentation des suggestions cruciale

Quelle modélisation?

Tour d'horizon

Automatiser des séquences d'actions

ou comment apprendre des macro-commande intelligentes…

Smartedit
(Lau et Domingos et al. 01, 02)

Automatiser des tâches répétitives d'édition

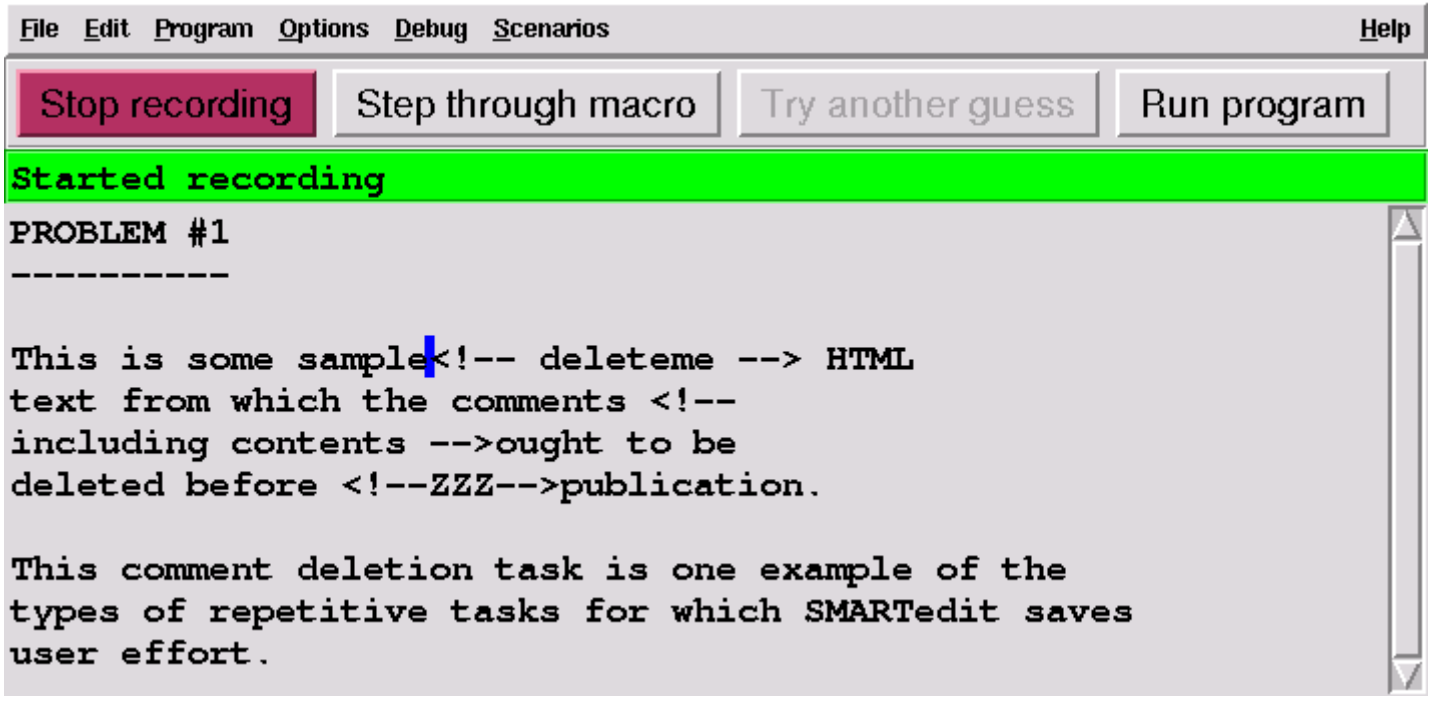
- L'utilisateur démarre un enregistreur de macro
- Accompli une tâche répétitive
- Stoppe l'enregistreur

- Apprend la procédure d'édition
- Pas de contre-exemple
- One-sided (most specific) version space pour inférer une macro intelligente
- Connaissance du domaine

- Prédit les actions de l'utilisateur

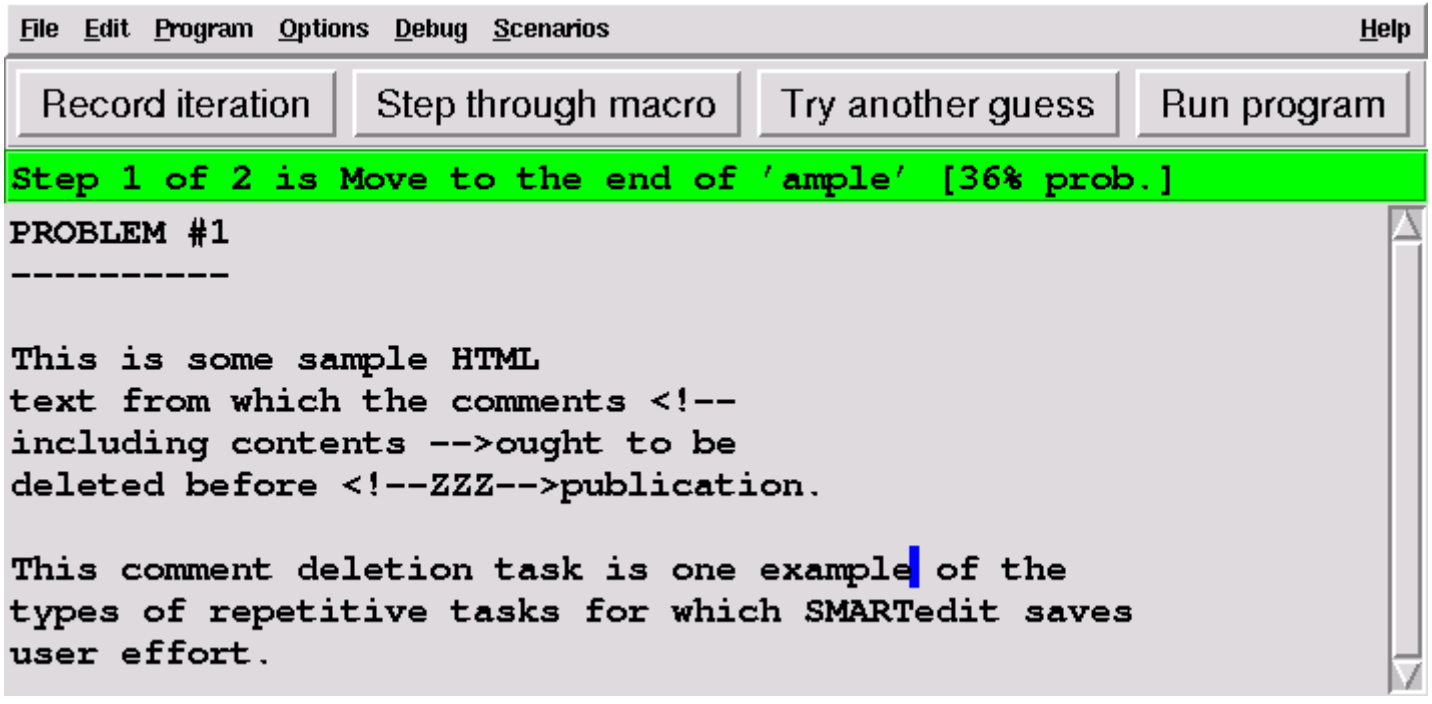
Smartedit : démo

L'utilisateur démarre l'enregistreur et supprime un commentaire…



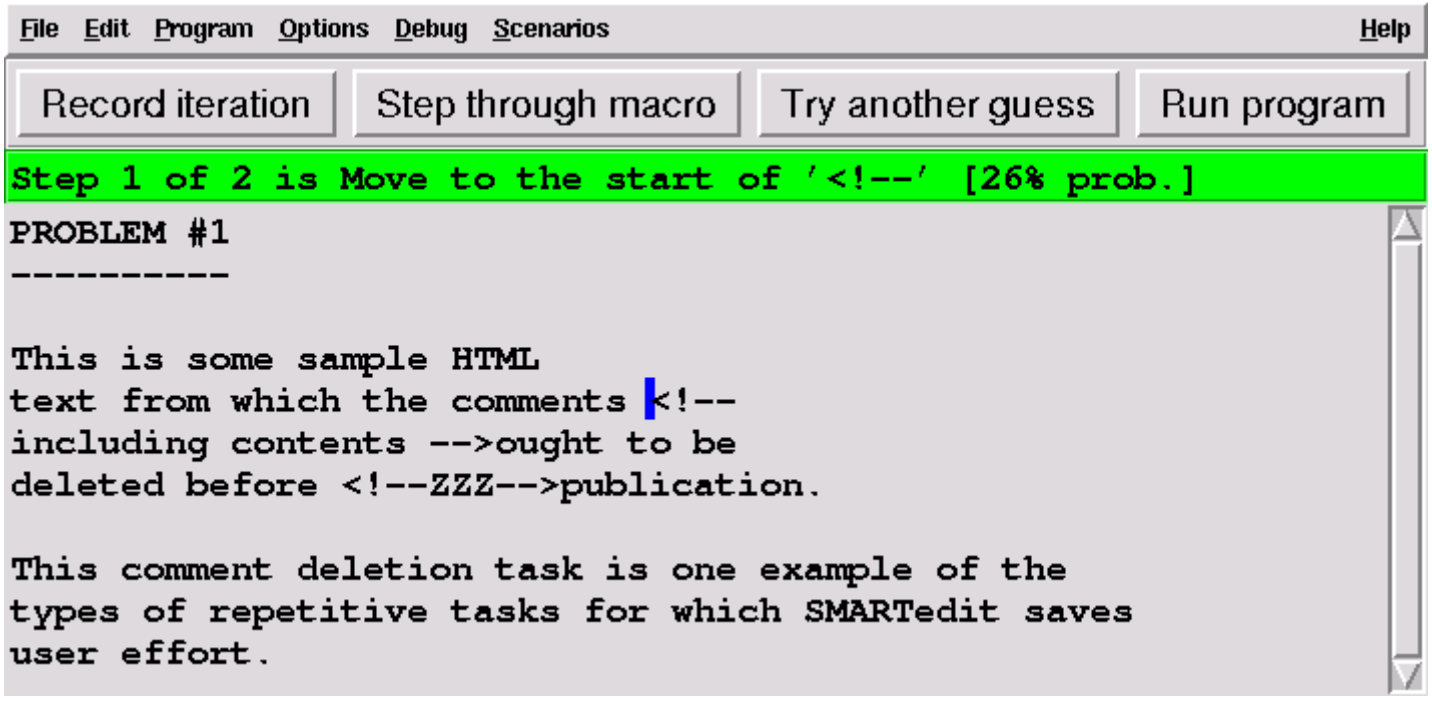
Smartedit : démo

L'utilisateur stoppe l'enregistreur. Suggestion incorrecte…

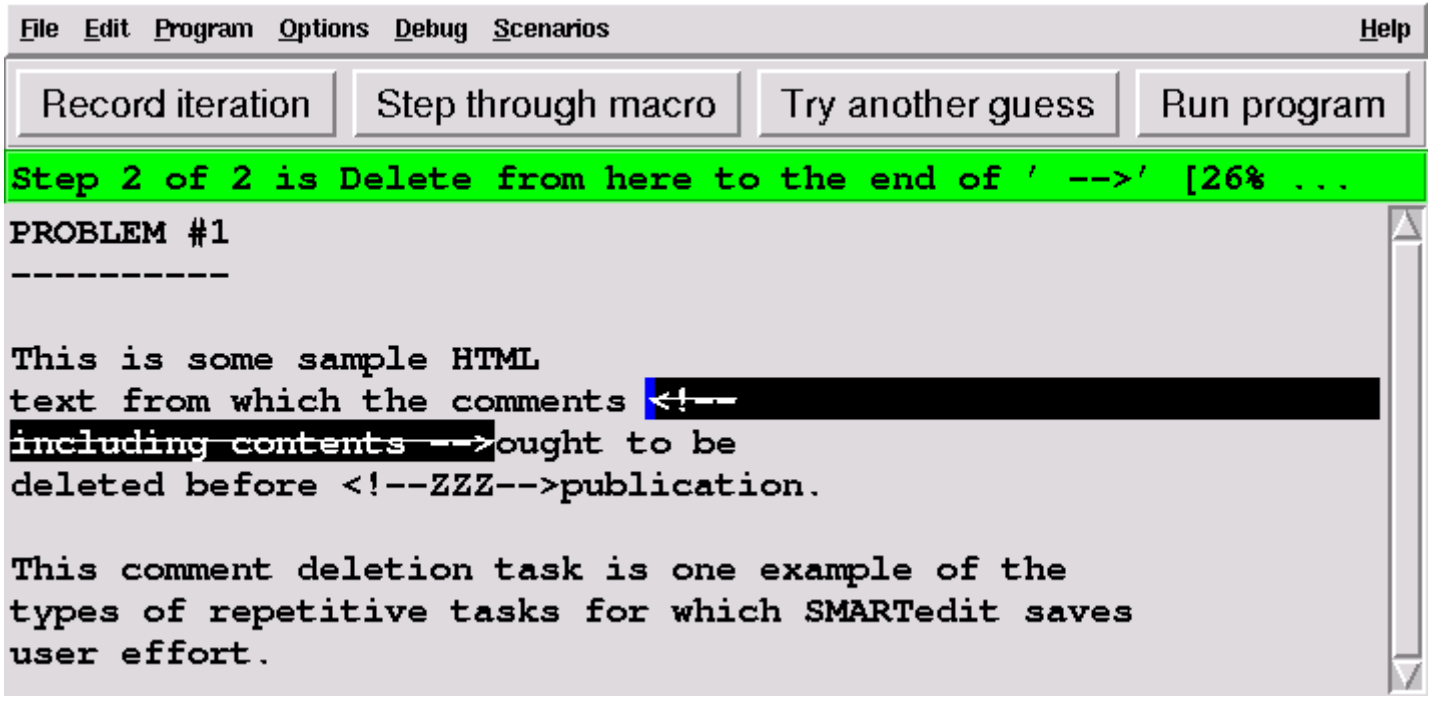


Smartedit : démo

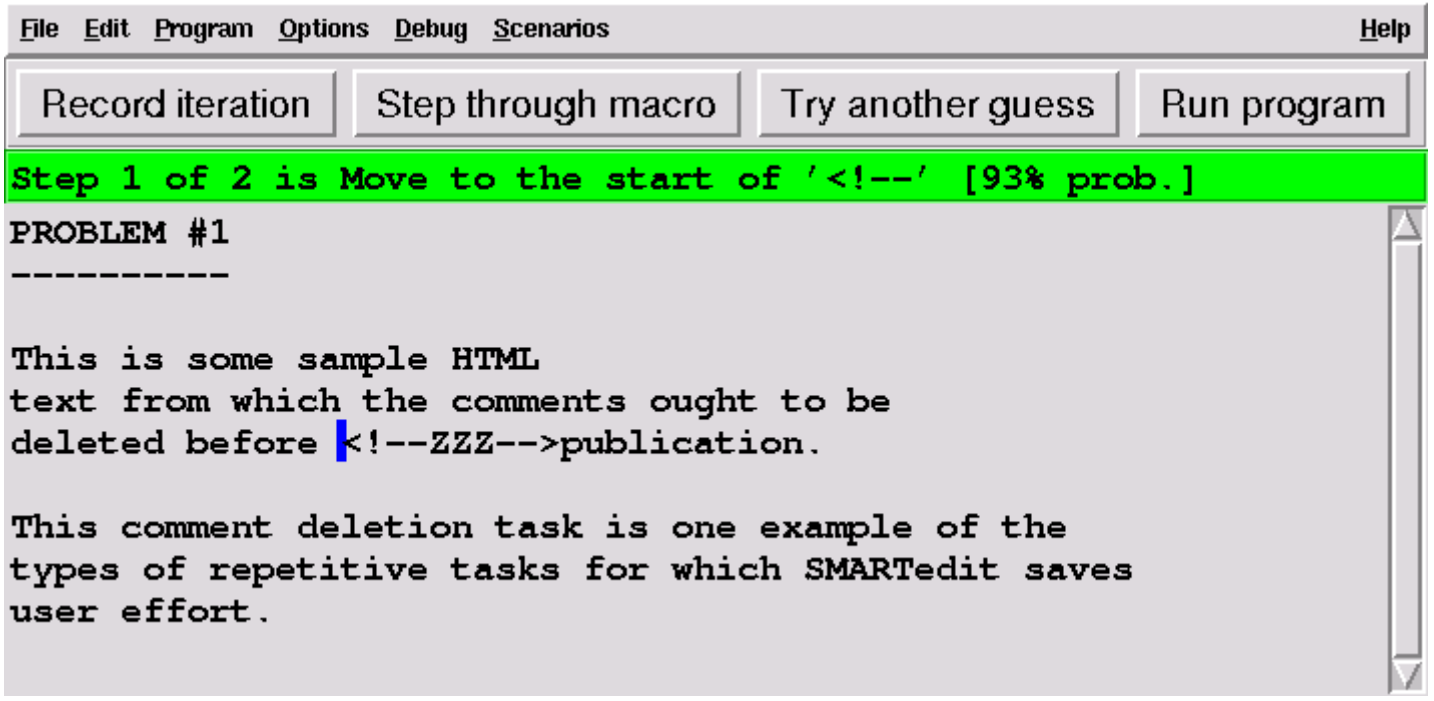
L'utilisateur a pressé « Try another guess». Suggestion correcte…



Smartedit : démo



Smartedit : démo



Nombre de clics réduit mais pas de gain de temps en moyenne

LAPIS
(Miller et Myers 02)

Multiple sélection/édition

- Multiple sélection : l'utilisateur sélectionne quelques exemples de texte à modifier
- LAPIS infère tous les autres exemples
  - Ajouter un exemple : sélectionner
  - Spécifier un contre-exemple : désélectionner
- Multiple édition : les modifications d'un exemple sont répercutées à tous les autres
- Algorithme conjonctif simple
- Performances
  - sélection 67%, édition 84%
  - Seuil gain de temps $\leq$ 9,3 entrées

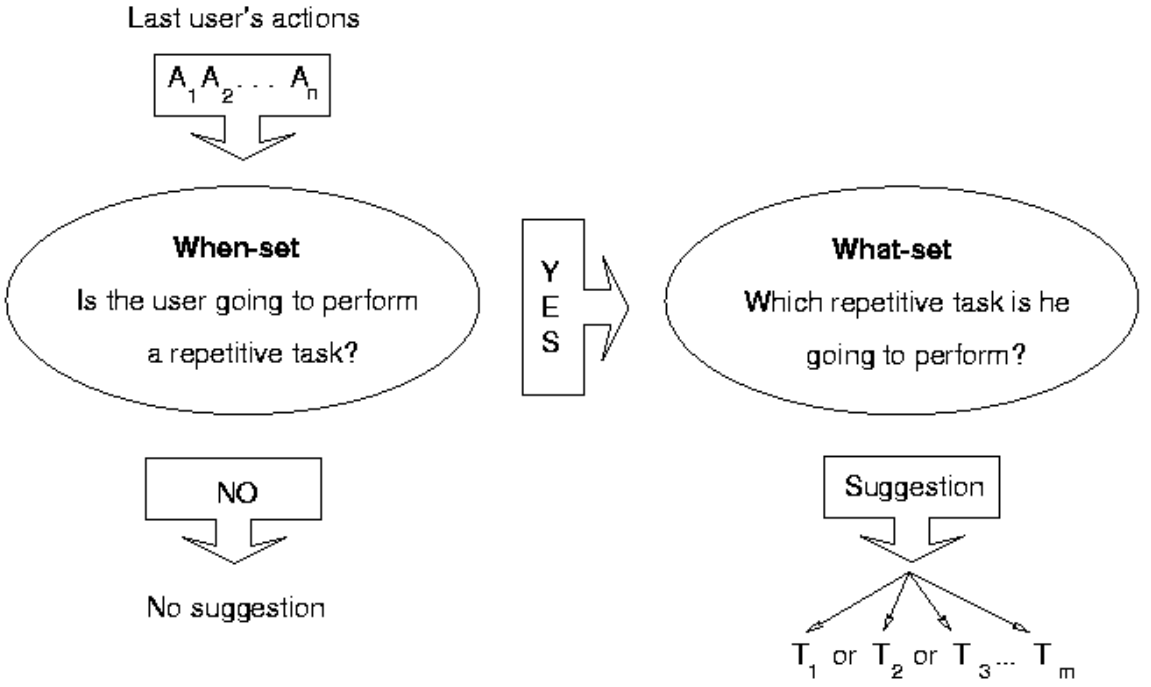
APE
(Ruvini et Dony 01)

Automatiser des tâches répétitives

- VisualWorks (Smalltalk) :
  - 70% des actions sont des répétitions
  - 20% des actions font partie d'une tâche répétitive
- S'affranchir de l'approche « enregistreur »
- Apprendre les habitudes de l'utilisateur =

  corrélations entre
  - Tâche répétitive : séquence d'actions, portions de code et erreurs syntaxiques répétitives, itérations
  - Situations

APE : deux algorithmes



C4.5 pour apprendre « quand »

Une variante de RISE pour apprendre « quelle tâche »

APE : démo

Démo

APE : performances

Suggestion correcte pour 80% des actions de l'utilisateur

- Anticipe 65% des tâches répétitives
- Suggestion excessive pour 15% des actions

Test utilisateur

- Présentation des suggestions cruciale
- Les utilisateurs ont appris à consulter APE au bon moment

Automatiser des séquences actions : résumé

Peu d'exemples

Pas de contre-exemple

Présentation des suggestions cruciale

Tour d'horizon

Autres applications

Autres applications

WebWatcher (Joachims et Mitchell et al. 97)

- TFIDF + renforcement + liens fréquemment suivis par les autres utilisateurs
- 48,9% de suggestions correctes

Navigation sur PALM (Anderson et Domingos et al. 01)

- Clustering + mélange de 10 modèles de Markov
- 40% de clics en moins
- Mélange de 10 MM meilleur que MM et mélange de NBC

Collaborative filtering (Goldberg et al. 92)

- recommander à un utilisateur des articles que des utilisateurs aux goûts similaires ont achetés (k-NN)
- Questions : nouveaux utilisateurs, produits; scalabilité

Une approche viable?

S'adapter?

Par apprentissage?

Tour d'horizon : modélisation, performances

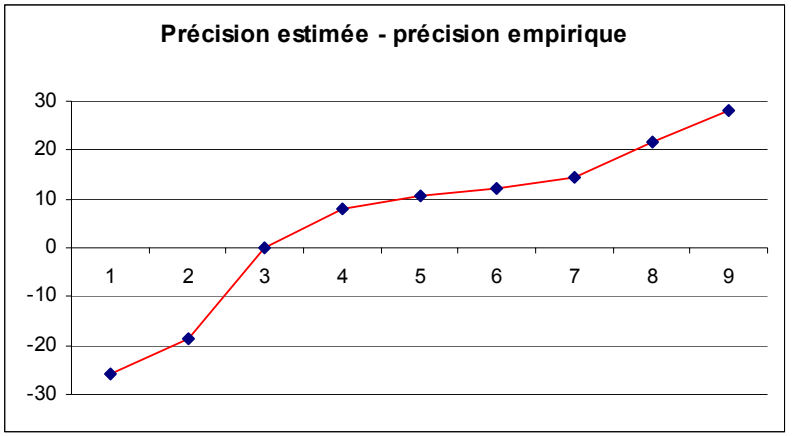
- Prédire des commandes textuelles
- Compléter des formulaires
- Filtrer des emails ou des news
- Automatiser des séquences d'actions
- Recommander des liens
- Recommander des produits

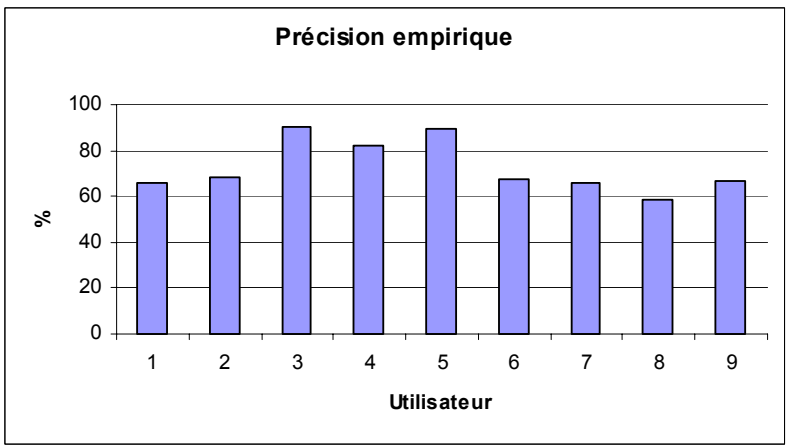
Une approche viable?

Résumé

Perspectives

Une approche viable?





Pour 3 suggestions

Test utilisateurs Smartlook:

Les utilisateurs sur-estiment les performances !

Résumé - Perspectives

S'adapter?

Par apprentissage?

Tour d'horizon : modélisation, performances

- Prédire des commandes textuelles
- Compléter des formulaires
- Filtrer des emails ou des news
- Automatiser des séquences d'actions
- Recommander des liens
- Recommander des produits

Une approche viable?

Résumé

Perspectives

Résumé : retour à Dieterich et al. 93

Objectifs :

1. Faciliter / accélérer l'utilisation
2. Présenter des info pertinentes

Mécanismes :

- Suggérer des commandes / des valeurs / des objets
- Créer des macro-commandes

Mode : (passif / actif)

- Automatique : CAP, APE, …
- WebWatcher : but de la recherche
- Smartedit, Familiar (Paynter et Witten 01) : enregistreur
- NewsDude : peut demander explications et critiques
- Route Advisor (Rogers et Langley et al 99) : quel critère est le plus important?

Résumé : retour à Dieterich et al. 93

Composant : modèle de l'utilisateur

- Individuel / collectif
- Sortie
  - Décision binaire : notification d'e-mails
  - Valeurs discrètes : formulaires
  - Score : la plupart des systèmes

Stratégie :

- Au cours des sessions

Cible :

- Utilisateur individuel

Résumé : leçons

Effet plafond :

- algorithmes non centraux (ad hoc / existant)
- La sophistication dégrade souvent les performances, lorsqu'elle aide, c'est modestement

100% de correction rarement possible

Mais les utilisateurs tolèrent des erreurs

Si les suggestions sont bien présentées

Résumé : leçons

Evaluation

- Taux d'erreur en prédiction
- Précision / rappel
- Gain en effort
- Test utilisateurs
- Comparaison à un humain

Résumé : une communauté

Hirsh, Domingos, Witten, Langley, Mitchell, Pazzani

IBM, Microsoft

Lieberman

Perspectives

Hypothèse de Markov

Arbitrer passé ancien / passé récent

Combiner plusieurs sources d'information

- Modèle individuel / général (Jameson et Wittig 01)
- Connaissance du domaine

Collaborative filtering

Méthodes meta :

- Active learning
- Cost sensitive learning

Représenter l'effet d'une suggestion

Documents

Cette présentation :

http://e-lab.bouygues.com

La thèse :

http://www.lirmm.fr/~ruvini

Programmation par l'exemple

Your Wish is My Command (MKP, Lieberman 01)