Object search

Gautier DI FOLCO

Janvier 2014

Table des matières

Résumé

Le but est de voir s'il est possible, à partir de n'importe quel site marchand, via *machine learning* d'en extraire les informations principales.

1 Méthodologie

Nous sommes partie sur une approche CRF et nous avons pour cela pris chaque noeud d'un document HTML (en ayant filtré un grand nombre de noeuds non-suceptible de nous intéresser pour les informations que nous cherchons, afin de limiter le nombre de noeuds inintéressant, qui rendraient l'apprentissage plus long) et nous avons établi une liste de *feature*s.

Nous avons manuellement marqué (via l'ajout d'un attribut *data-tess-label*) une partie (10 pages) de notre jeu de données (il comporte 50 pages par site, ces sites étant amazon.fr carrefour, ldlc, fnac et rueducommerce) afin de nous en servir comme base d'apprentissage pour notre logiciel de CRF (*CRFSuite*).

1.1 Features extraites

parProd Le nombre de parents ayant une classe contenant le mot "prod"
ancProd Le nombre d'ancêtres ayant une classe contenant le mot "prod"
ancDesc Le nombre d'ancêtres ayant une classe contenant le mot "desc"
selfProd Le nombre de classes ayant "prod" dans son nom pour le noeud courant
selfDesc Le nombre de classes ayant "desc" dans son nom pour le noeud courant

selfCurr Le nombre d'occurences d'un symbole de monnaie dans le texte du noeud courant
contDesc Le nombre d'occurences de "desc" dans le texte du noeud courant
selfEl Le nom du noeud courant
parEl Le nom du noeud parent
selfIP Le contenu de l'attribut *itemprop*
selfClass La classe courante
parClass La classe du noeud parent
selfDepth Le nombre d'ancêtres
selfChilds Le nombre de déscendants
inHn Le noeud courant ou un de ses ancêtre est-il un noeud de type h1, h2 ou h3

1.2 Données recherchées

Nous avons tenté de chercher la description, le titre et le prix de chaque page.

Dans cet objectif, notre convertisseur de HTML en *dataset CRFSuite* va, pour chaque noeuds non-filtré de la page, chercher si ce noeud a l'attribut *data-tess-label* si c'est le cas, le type de l'enregistrement aura cette valeur, si non il aura comme type "other".

1.3 Méthodologie

Dans un premier temps nous récupérons un jeu de 50 pages pour 5 sites, nous en marquons manuellement 10.

Nous convertissons ensuite ces deux jeux (l'original de 50 pages et le marqué de 10 pages) en format CRFSuite.

Puis nous lançons la procédure d'apprentissage, une fois les modèles générés nous les appliquons sur les pages non-marquées au format CRFSuite.

Ensuite nous convertissons le résultat en page HTML.

Nous récupérons de plus des statistiques sur les phases d'apprentissage et de marquage.

1.3.1 Première tentative : inter-site

Lors de l'apprentissage nous avons générés les 120 combinaisons possibles de sites (afin de voir si l'ordre des jeux de données influence l'apprentissage) puis nous avons retirer un jeu (le dernier site de la liste) et nous avons fait un apprentissage sur les jeux restants.

Puis nous appliquons chaque modèles aux pages du jeu restant.

Nos observations sont les suivantes :

— L'ordre des jeux de données à une influence sur le temps d'apprentissage
— L'apprentissage est très long (entre 20h et 6 jours)
— Les modèles générés sont inefficaces (aucun marquage n'a remonté une information recherchée)

Nous en concluons qu'il faut, avant de poursuivre les investigations plus loin, valider que CRFSuite est fonctionnel.

1.3.2 Deuxième tentative : site par site

Les pages de chaque site étant générés dynamiquement à partir d'un gabarit fixe CRFSuite devrait pouvoir s'y retrouver plus facilement si chaque site à un apprentissage et un marquage isolé.

Malheureusement ce n'est pas le cas, les temps d'apprentissages sont encore long (de quelques heures à quelques jours) et les marquages sont encore inefficaces.

Nous en déduisons donc que soit le nombre de noeuds "other" est trop important par rapport aux noeuds intéressants, soit les features sont mal choisies/insuffisantes.

1.3.3 Troisième tentative : évaluation des *feature*s

Le but est de sélectionner les *feature*s les plus discriminantes.
$Soit \Phi_i une$*feature*..
Pour $val \in \{titre, description, prix\}$.
Pour $i = 1..n$ où n est le nombre de *feature*s.
Soit

$$\Delta^{val}_{\Phi_i} = \frac{|moyenneValeurs(\Phi_i), elementsP) - moyenneValeurs(\Phi_i), elementsN)|}{maxValeur(\Phi_i) - minValeur(\Phi_i)}$$

où $elementsP$ est l'ensemble des noeuds marqués par val et $elementsN$ les autres.

On choisit les K *feature*s les plus driscriminantes (qui ont un $\Delta^{val}_{\Phi_i}$ le plus proche de 1 possible).

On va ensuite chercher une valeur seuille pour laquelle le discriminant est pertinent.

Voici pour chaque site les résultats :

amazon Voici les *feature*s par ordre décroissant de pertinence :

- title inHn=0.9815057868989381 selfDepth=0.15146107124391342 ancProd=0.03328958358191147
- price ancProd=0.9667104164180885 selfDepth=0.10568178589894371 selfCurr=0.0189714831165732
- description selfProd=0.990096647178141 selfDepth=0.23241345219629436 ancProd=0.03328958358191147

Les *feature*s sélectionnées pour la phase d'apprentissage sont donc : inHn selfDepth ancProd selfCurr selfProd

carrefour Voici les *feature*s par ordre décroissant de pertinence :

- title inHn=0.962248322147651 ancProd=0.9253355704697986 selfDepth=0.12138213087248317
- img ancProd=0.07466442953020135 selfDepth=0.058882130872483174 inHn=0.037751677852348994
- price selfDepth=0.25263213087248315 ancProd=0.07466442953020135 inHn=0.037751677852348994
- description ancProd=0.07466442953020135 inHn=0.037751677852348994 ancDesc=0.03271812080536913

Les *feature*s sélectionnées pour la phase d'apprentissage sont donc : inHn ancProd selfDepth ancDesc

fnac Voici les *feature*s par ordre décroissant de pertinence :

- img selfDepth=0.1394904349903582 inHn=0.047577603713373946 ancProd=0.035030461270670145
- title inHn=0.9524223962866261 selfDepth=0.11810006065880743 ancProd=0.035030461270670145
- price selfDepth=0.12879524782458282 inHn=0.047577603713373946 ancProd=0.035030461270670145
- description selfDepth=0.29457064889410156 inHn=0.047577603713373946 ancProd=0.035030461270670145

Les *feature*s sélectionnées pour la phase d'apprentissage sont donc : selfDepth inHn ancProd

ldlc Voici les *feature*s par ordre décroissant de pertinence :

- price selfCurr=0.9768084779603747 parProd=0.14252802948855783 selfDepth=0.10417946551988944
- title inHn=0.980187375211181 parProd=0.14252802948855783 selfDepth=0.08332053448011056
- img selfDepth=0.14582053448011056 parProd=0.14252802948855783 ancProd=0.06839707162238266
- description selfDesc=0.9906312394409461 ancDesc=0.6569395382173757 parProd=0.14252802948855783

Les *feature*s sélectionnées pour la phase d'apprentissage sont donc : selfCurr parProd selfDepth inHn ancProd selfDesc ancDesc

rueducommerce Voici les *feature*s par ordre décroissant de pertinence :

- img selfDepth=0.13216042606821976 selfCurr=0.0349432857665569 ancDesc=0.023783388218075376
- title inHn=0.9853640687888767 selfCurr=0.0349432857665569 ancDesc=0.023783388218075376
- price selfDepth=0.25672846282066913 selfCurr=0.0349432857665569 ancDesc=0.023783388218075376
- description ancDesc=0.9762166117819246 selfDepth=0.1877159816237753 selfCurr=0.0349432857665569

Les *feature*s sélectionnées pour la phase d'apprentissage sont donc : selfDepth selfCurr ancDesc inHn