



Easter-eggs (/)

Études de cas (-Études-de-cas-) / Développement applicatif (-Développement-applicatif-)

Spécialiste GNU/LINUX

/ Recherche approximative avec Double Métaphone et distance de Levenshtein

Membre du réseau **Libre-entreprise** (<http://www.libre-entreprise.com>)

Recherche approximative avec Double Métaphone et distance de Levenshtein

Présentation

La recherche approximative, ou Fuzzy search en anglais, concerne les recherches textuelles pour lesquelles une orthographe inexacte est permise.

Il existe des outils d'indexation et de recherche de données de type Apache Lucene [1], mais le besoin du client reste centré sur la recherche approximative donc nous développons une solution sur mesure.

Cet article présente une manière simple et élégante d'implémenter cette recherche approximative.

Fonctionnalités

Cet article utilise un exemple d'utilisation tiré du projet E-Nota [2] de notre client Médiamétrie [3].

Ce site est accessible aux utilisateurs abonnés au service.


Le site web E-Nota affiche, entre autres, des informations sur des programmes télévisés stockés en base de données. L'utilisateur accède aux fiches des programmes à l'aide d'un moteur de recherche.



Quick Search

Search for:

<input checked="" type="radio"/> Program title	<input type="text"/>	All of the words ▾
<input type="radio"/> Full text	<input type="text"/>	All of the words ▾

☐ Only show programs with video preview 

Search

Les titres des programmes étant stockés en anglais ainsi que dans leur langue d'origine, l'utilisateur se trompe fréquemment dans l'orthographe.

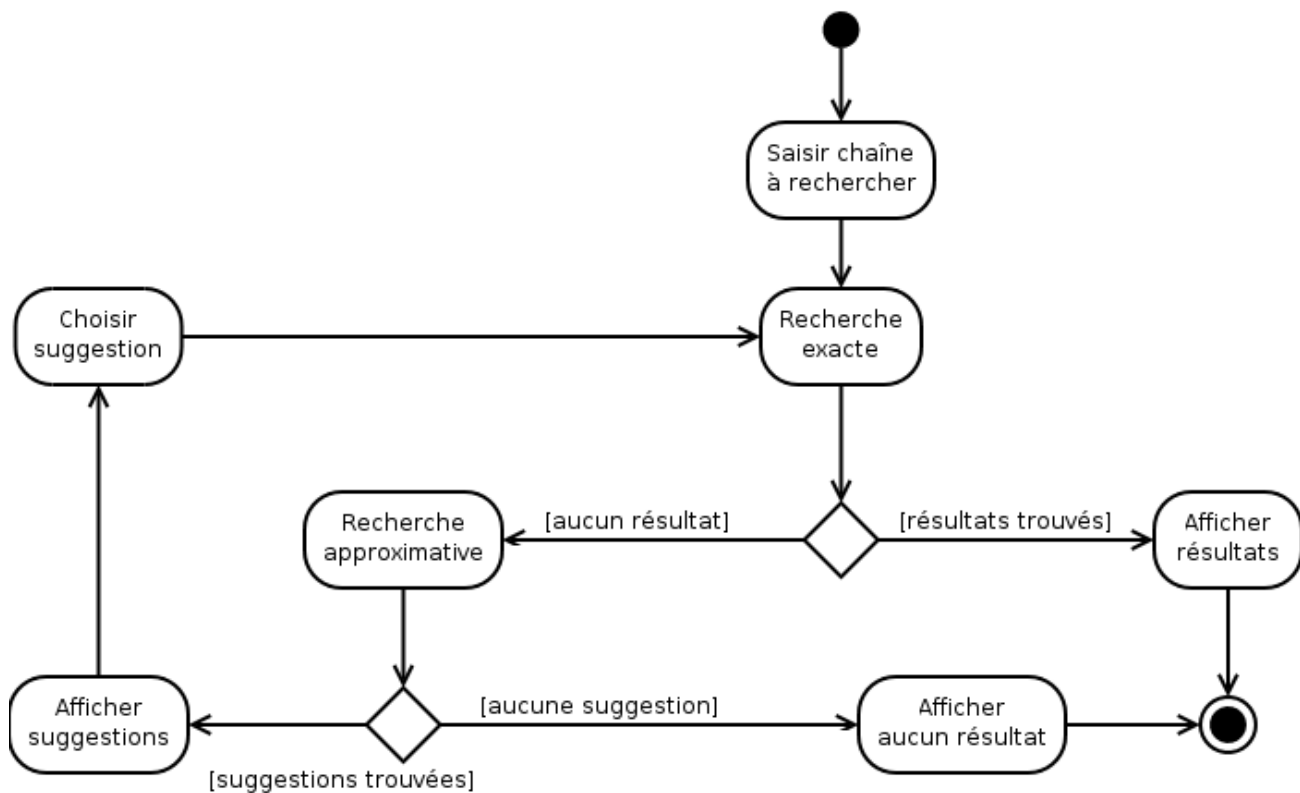
Le moteur de recherche doit être capable de trouver les bons titres de programmes, même si l'orthographe n'est pas la bonne, et cela de la façon la plus permissive possible.

Par exemple, si l'utilisateur saisit « Desesperate housevifs » (mauvaise orthographe), le moteur de recherche lui suggère « Desperate Housewives » comme alternative.

Dans un premier temps, les programmes sont recherchés selon l'orthographe exacte demandée par

l'utilisateur. Si aucun résultat n'est trouvé, l'algorithme de recherche approximative est utilisé afin d'élargir le champ des résultats possibles.

Voici le diagramme de navigation du moteur de recherche :



Codage phonétique

Présentation

La recherche approximative est résolue en utilisant le codage phonétique des titres des programmes.

Le titre de chaque programme est codé par une chaîne de caractères appelée clé phonétique.

La taille de la base de données est trop grande (environ 10000 programmes) pour calculer à la volée les clés phonétiques des titres : elles sont stockées dans une colonne de la table des programmes.

Il existe plusieurs algorithmes de calcul pour obtenir ces clés phonétiques. Nous présentons le Soundex et le Double métaphone.

Distance de Levenshtein

La distance de Levenshtein est un algorithme qui mesure la similarité entre deux chaînes de caractères.

Cet algorithme est utilisé dans la suite de l'article.

Il retourne le nombre de modifications à effectuer pour passer d'une chaîne à l'autre.

Les modifications sont l'ajout, la suppression ou la modification d'un caractère.

Soundex

L'algorithme du Soundex retourne une clé phonétique constituée d'une lettre et de quatre chiffres en fonction d'une chaîne de caractères fournie en entrée.

La première lettre est la première de la chaîne de caractères à coder. Les trois chiffres correspondent aux trois premières consonnes de la chaîne.

Cet algorithme historique est souvent utilisé mais n'est pas le plus efficace.



Par exemple, les chaînes « strictly », « Strictly come dancing » et « Strict in dancing » ont le même code : S362. Cela induit un manque de précision et donc du bruit dans les résultats de la recherche.

Pour effectuer la recherche, il faut :

1. trier tous les programmes en fonction de la distance de Levenshtein entre le codage du titre et le codage de la chaîne recherchée
2. sélectionner les X premiers programmes
3. trier de nouveau les résultats en fonction de la distance de Levenshtein entre le titre du programme et la chaîne recherchée
4. afficher les Y premiers programmes sous forme de suggestions de recherche

X et Y sont des constantes. Les valeurs $Y = 10$ et $X = Y + 30$ sont des valeurs intéressantes. Il est nécessaire de paramétrer X plus grand que Y pour diminuer le bruit dans les résultats de la recherche.

Malgré cela, les résultats présentés ne sont pas assez pertinents en raison du bruit engendré par l'algorithme. Le fait que l'algorithme du Soundex se limite aux trois premières consonnes est trop limitant. Il nous faut donc un autre algorithme qui offre un codage phonétique plus représentatif de la chaîne à coder.

Double métaphone

Dérivé de l'algorithme Métaphone, il est appelé « double » car il propose un codage principal et un alternatif pour une même chaîne donnée en entrée. Cela permet plus de souplesse quand aux différentes prononciations qui peuvent exister en fonction des pays.

Cependant nous n'utiliserons que le codage principal.

L'algorithme du double métaphone simplifie à l'extrême la chaîne donnée en entrée en lui retirant ses voyelles et en ramenant les consonnes similaires à une consonne de référence.

Par exemple, le titre « Desperate housewives » a pour codage « TSPRTSFS », la mauvaise orthographe « Desesperate housevif » est codée « TSSPRTSFF » et le mot « desperate » est codé « TSPRT ». Cela permet plus de finesse par rapport à l'algorithme du Soundex.

Les codages phonétiques étant différents, il est nécessaire de les comparer en utilisant non pas l'égalité stricte mais la distance de Levenshtein.

La méthode de recherche devient :

1. sélectionner tous les programmes dont la distance de Levenshtein entre le codage du titre et le codage de la chaîne recherchée est inférieure à L
2. trier ces programmes en fonction de la distance de Levenshtein entre les titres eux-mêmes et la chaîne recherchée
3. afficher les X premiers programmes sous forme de suggestions de recherche

Les valeurs $L = 2$ et $X = 10$ sont des valeurs intéressantes.

Cette fois-ci, il est possible de sélectionner directement dès la deuxième étape un nombre restreint de programmes à afficher. Cela engendre un gain en performances et en pertinence des résultats.

Pour pousser au maximum l'exemple, la recherche « strictli kum tenzy » renvoie bien « Strictly come dancing » et la recherche « desperat ouzviv » renvoie bien « Desperate Housewives ».

Implantation

Nous utilisons PHP version 5.2.0-8+etch11 et PostgreSQL version 8.1.11-0etch1 sous Debian Etch (stable).

La table « programs » stocke les programmes. Elle contient une colonne pour le titre et une colonne pour le codage double metaphone du titre.

```
> SELECT * FROM programs;
```

title	title_double_metaphone
Desperate housewives	TSPRTSFS
Strictly come dancing	STRKTLKMTNSNK
Lost	LST

L'utilisateur saisit la chaîne recherchée dans un formulaire d'une page HTML. La variable s'appelle `$search_text`.

Le calcul de la clé phonétique double métaphone utilise la fonction PHP `double_metaphone()` non disponible en standard. Elle est fournie par l'extension PECL `doublemetaphone` [4].

De plus nous avons besoin d'installer la fonction « levenshtein » pour PostgreSQL :

```
root# apt-get install postgresql-contrib-8.1
user$ psql my_database < /usr/share/postgresql/8.1/contrib/fuzzystrmatch.sql
```

La procédure de recherche encode la chaîne à rechercher :

```
$search_text_dm = double_metaphone($search_text);
$search_text_dm = $search_text_dm[0];
```

La fonction `double_metaphone` renvoie un tableau à deux éléments : le codage principal et alternatif. Nous n'utilisons que le codage principal.

Puis recherche les programmes grâce à la requête SQL suivante :

```
SELECT DISTINCT title, LEVENSHTEIN(LOWER($search_text), LOWER(title)) AS distance
FROM programs
WHERE LEVENSHTEIN($search_text_dm, title_double_metaphone) < 2
ORDER BY distance
LIMIT 10;
```



Évolutions possibles

Il existe encore une limitation : le codage en double métaphone est effectué sur le titre du programme en entier en non pas mot par mot. Si l'utilisateur saisit une partie du titre mal orthographiée, les résultats ne seront pas assez pertinents.

Par exemple, s'il saisit « housevif » au lieu de « housewives », les résultats retournés seront des programmes dont le titre commence par « house », « as », « heist », etc. tandis que l'utilisateur s'attendait à « Desperate housewives ».

Pour résoudre ce problème, on pourrait coder séparément les mots en double métaphone dans une table à part. Dans ce cas, une table d'indirection serait utilisée pour effectuer la recherche.

Puis l'algorithme de recherche travaillerait mot par mot.

Il est envisageable de porter cette méthode de recherche approximative comme plugin pour un framework web.

Le framework Symfony [5] propose déjà un plugin pour Apache Lucene, un moteur de recherche générique.

Conclusion

Les résultats obtenus sont très satisfaisants car la tolérance aux erreurs est grande et les résultats

pertinents.

-
- [1] Apache Lucene, <http://lucene.apache.org> (<http://lucene.apache.org>)
 - [2] New On The Air, <http://www.e-nota.com> (<http://www.e-nota.com>)
 - [3] Médiamétrie, <http://www.mediametrie.fr> (<http://www.mediametrie.fr>)
 - [4] Extension PECL doublemetaphone, <http://pecl.php.net/package/doublem...> (<http://pecl.php.net/package/doublemetaphone>)
 - [5] Framework Symfony <http://www.symfony-project.org> (<http://www.symfony-project.org>)



Easter-eggs

44/46, rue de l'Ouest
75014 Paris France

☎ +33 (1) 43 35 00 37 (tel:+33143350037)

💬 Nous contacter (Nous-contacter)

Études de cas

- 📖 Dovecot : fonctionnalités IMAP avancées (Dovecot-fonctionnalites-IMAP)
- 📖 Optimisation des applications X11 avec XCB (Optimisation-des-applications-X11)
- 📖 Intégration d'un langage de programmation dans une application avec Lua (Integration-d-un-langage-de)
- 📖 Système d'impression PDF « RESTful » pour le Web (Systeme-d-impression-PDF-RESTful)
- 📖 Retour sur l'optimisation des performances d'une base de données PostgreSQL (Retour-sur-l-optimisation-des)



✚ Plus d'études (/ -Etudes-de-cas-)

À lire également

- 🔗 Le blog des salariés (<http://blog.easter-eggs.org/>)
- 🔗 Planet Easter-eggs (<http://planet.easter-eggs.org/>)

Le coin des salariés

- 📰 Sans titre (par le 31/12/2015) ()
- 📰 Des nouvelles du Roulard ! (par Emmanuel Saracco le 17/10/2015) (<https://www.leroulard.com/>)
- 📰 Vive cyclododo ! (par Emmanuel Saracco le 31/08/2015) (<http://www.cyclododo.com/>)
- 📰 Sortie de wbmclamav 0.16 (par Emmanuel Saracco le 16/07/2015) (<http://wbmclamav.labs.libre-entreprise.org/>)
- 📰 LeRoulard - Plateforme d'Assistance à la Randonnée (par Emmanuel Saracco le 11/07/2015) (<https://www.leroulard.com/>)

© 1997-2014 Easter-eggs, tous droits réservés.

