L'expansion de requête est-elle à la base de Hummingbird ?

Retour au sommaire de la lettre

Domaine :	Recherche	**Référencement**
Niveau :	Pour tous	**Avancé**

Google a annoncé fin septembre le lancement, un mois plus tôt, de son nouvel algorithme baptisé Hummingbird. Cependant, peu d'impacts ont été identifiés dans les résultats du moteur de recherche suite à cette annonce alors que Google indiquait que 90% des recherches étaient touchées. Mais il semblerait que Hummingbird s'attache plus à désambigüiser les requêtes de l'internaute plutôt qu'à améliorer la formule de classement proprement dite. Comment cela peut-il se faire ? La plupart du temps en passant par des techniques d'"expansion de requête". Voici quelques pistes de réflexion et une revue des différentes façons d'effectuer ce traitement...

Le 26 septembre 2013 était une date importante pour Google, qui fêtait son quinzième anniversaire dans le garage des origines. L'événement était un peu anecdotique pour ceux qui ne sont pas touchés par le coté nostalgique et fondateur de la chose, mais ce qu'il faut retenir est une annonce liée au fonctionnement du moteur : l'utilisation d'un nouvel algorithme, appelé « Hummingbird », pour améliorer l'expérience utilisateur, en particulier au niveau de la requête.

Dans cet article, nous n'allons pas faire de trop nombreuses spéculations sur ce qu'est précisément ce « colibri », nous ne sommes pas dans le secret de Google, mais plutôt prendre pour acquis le fait qu'il s'agit principalement d'un algorithme de « query expansion » et passer en revue tout ce que cela implique. C'est-à-dire que l'on va utiliser Hummingbird comme prétexte pour expliquer ce qu'un moteur moderne peut faire comme travail au niveau de la requête.

Query expansion ? Une définition rapide.

Le problème avec les requêtes qu'un internaute utilise pour interagir avec un moteur de recherche, c'est qu'elles sont courtes (entre 3 et 5 mots généralement), qu'elles sont ambigües, et enfin qu'elles peuvent être trop spécifiques. L'ambigüité peut provenir de l'utilisation de mots dont le sens est variable selon le contexte (orange, jaguar, avocat, etc.), phénomène qui peut être amplifié par des soucis grammatico-syntaxique (« l'homme, ferme, la porte » est différent de « l'homme ferme la porte », « on mange, les enfants » est différent de « on mange les enfants », etc.).

Le problème de spécificité est tout autre : il s'agit des cas où la formulation de la requête utilise des termes qui ne sont pas ceux qu'on retrouve dans les documents qui sont pertinents. Par exemple, l'utilisation de « dérivés sur événement de crédit » au lieu de « contrat de protection financière » ne permet pas à l'internaute de trouver facilement ce qu'il recherche en réalité.

Pour lutter contre ces problèmes liés à la requête, on peut utiliser des techniques de « query expansion ». Il n'existe pas de traduction française consacrée, mais il s'agit de reformulation de requêtes plus que d'expansion de requêtes. Le « expansion » du terme anglais se rapportant plus à une expansion au sens sémantique qu'à une mystérieuse « extension » de la requête.

Il existe plusieurs types de techniques pour faire cette reformulation, mais les grands classiques poursuivent les buts suivants :

- **Reformulation** d'une requête en une requête **structurellement identique**, mais sur des mots différents (des synonymes la plupart du temps). « Je me suis fait carjacker » devient ainsi « on m'a volé l'auto ».

- **Correction orthographique** de la requête. Là où l'internaute écrira « mailleure commenterre skyblogs », le moteur comprendra « meilleurs commentaires skyblog ».

- **Reformulation par relation conceptuelle**. Nous expliquons plus en détail cette technique plus loin, mais il s'agit basiquement de remplacer des termes par d'autres, *via* des relations connues entre les concepts sous-jacents. On transformera ainsi « Dans quels films joue l'actrice de Spiderman ? » par « Quels sont les films dont Kirsten Dunst est une actrice ? ».

- **Reformulation par proximité, statistiques**, etc. Pour lever des ambigüités, on va utiliser une information contextuelle. Par exemple, le mot « orange » dans la requête « SAV orange » va correspondre à l'opérateur de téléphonie, tandis que dans la requête « orange pressée », il s'agira du fruit.

- **Correction des facteurs de pondération** de certains termes. *A priori*, il s'agit d'une technique peu utilisée, mais on pourrait imaginer que dans certains contextes, un mot bénéficie d'une pondération supérieure.

- **Reformulation par modification morphologique**. Il s'agit ici de transformer les mots en des versions morphologiquement proches. Il peut s'agir de morphologie flexionnelle (deux formes d'un même verbe) ou lexicale (rappeur comme extension du mot rap). Ici on est dans une approche linguistique, difficile à mettre en œuvre pour un moteur.

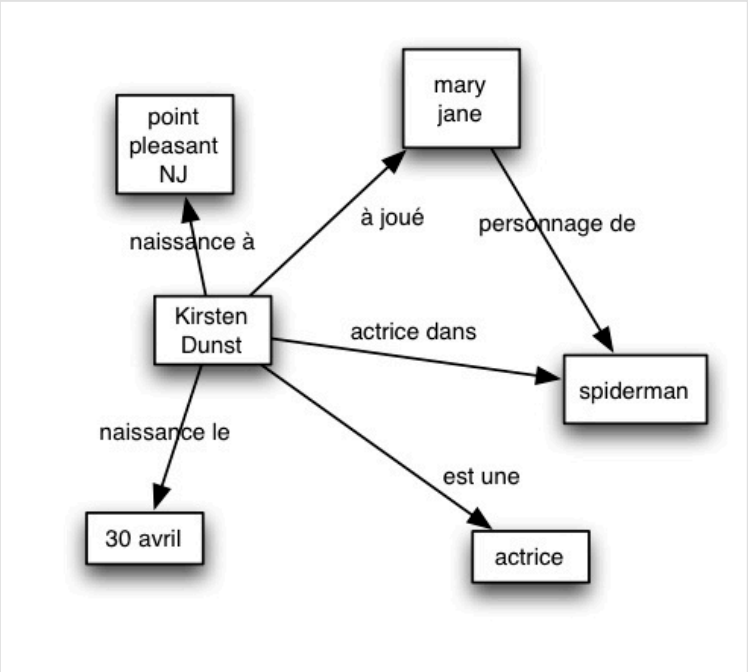
Une fois cette introduction faite, nous allons passer en revue trois procédés de reformulation de requête qui sont « google-compatible ». Google-compatible, cela veut dire qu'il existe un article ou un brevet du géant de Mountain View qui décrit la technique, ou bien que Google possède la machinerie qui permet de la mettre en place (et qu'il serait vraiment dommage de ne pas le faire).

Nous allons commencer par le plus prospectif : l'utilisation du knowledge graph.

Knowledge graph et reformulation de requête

Le but de cet article n'est pas de s'étendre sur la façon dont est fabriqué le Knowledge graph, mais plutôt de savoir comment on peut l'utiliser pour la reformulation de requête. Nous allons donc simplement rappeler ici que le Knowledge graph est une représentation graphique d'une ontologie, c'est-à-dire un modèle de données qui représente les concepts et leurs relations. On part d'objets de base (des entités telles que Saturne, Paris, Voiture, etc.) sur lesquels on va expliciter des attributs (caractéristiques d'un objet comme par exemple la taille, le poids, le sexe, etc.). On va ensuite définir des relations entre objets (« Alice est une femme », « la voiture est rouge », etc.).

Par exemple, la figure suivante présente le Knowledge graph issu du traitement des premières lignes de la fiche wikipedia de Kirsten Dunst :



L'idée même de reformulation à l'aide du knowledge graph devient alors triviale. Imaginons que l'internaute saisisse « Où est née l'actrice qui a joué Mary Jane dans Spiderman », le moteur va alors chercher la chaîne de relations « X – joue – Mary Jane – personnage de – Spiderman » et d'autres chaines similaires « Mary Jane – est joué par – X – dans – Spiderman », etc. Le X sera le terme de reformulation. Dans notre exemple il s'agit de Kirsten Dunst, et la requête sera donc reformulée « Où est née Kirsten Dunst », qui est une requête pour laquelle le moteur sait parfaitement renvoyer des résultats pertinents.

Est-ce que cette technique est mise en place à l'heure actuelle ? Nous ne pouvons le dire avec certitude, mais cela paraît tellement simple que ce serait étonnant que cela ne le soit pas, d'autant plus que dès 2009, Amit Singhal annonçait que le principal but du Knowledge Graph était de lever les ambigüités au niveau de l'interface avec l'utilisateur.

Reformuler en trouvant des mots de substitutions

Avec cette technique, on entre dans ce que Google sait faire puisqu'il s'agit d'un procédé décrit dans un brevet deposé par la société :

Evaluation of substitute terms
Inventeurs : Daisuke Ikeda and Ke Yang
Plus d'infos : http://goo.gl/YQX0ta

La demande concernant ce brevet date de 2012, et a été acceptée en 2013. Là encore, rien de vraiment compliqué, il s'agit simplement de déterminer des termes de remplacement pour élargir le spectre de la recherche en terme de documents. En effet, un problème pour Google est que l'internaute va choisir un mot pour caractériser sa recherche parmi de nombreux mots possibles : on choisit d'écrire « voiture », mais on aurait pu mettre « automobile », « bagnole », « caisse », etc.

Ici, on va donc chercher des termes qui sont des bons substituts, de manière statistique. Il ne s'agit pas à proprement parler de synonymes mais plutôt de données d'usage , récupérées par Google grâce à l'analyse des requêtes de ses utilisateurs.

L'idée va être de regarder quels sont les termes qui apparaissent comme étant très présents dans les résultats pour la requête de base, puis de considérer ces termes comme

étant de possibles substituts, ce que l'on peut vérifier statistiquement en regardant les requêtes qui utilisent ces termes.

Voyons un exemple pour mieux comprendre. Imaginons que l'internaute tape « sport automobile ». Dans les premiers résultats, on voit apparaître les mots « voiture », « F1 », « rallye » et le moteur va donc vérifier (par exemple) si les requêtes reformulées « voiture de sport », « F1 » et « rallye automobile » sont raisonnables. Statistiquement il verra rapidement que « voiture de sport » n'est pas correcte, mais que les deux autres sont utilisables.

On est ici sur une technique déposée par Google et très probablement utilisée.

Reformuler en utilisant la co-occurrence

Il s'agit de la technique qui est présentée dans ce que Bill Slawski a appelé « le brevet Hummingbird » (http://www.seobythesea.com/2013/09/google-hummingbird-patent/) et dont le nom est « Synonym identification based on co-occurring terms » (http://goo.gl/n1q1hK).

L'idée d'utiliser la co-occurrence n'est pas neuve chez Google, un brevet sur le même sujet ou presque a déjà été déposé en 2005. Et au delà, l'idée est très classique en *information retrieval*, avec des brevets par Xerox et NEC avant 2000, et même un article scientifique datant de 1977 sur le sujet de la co-occurrence.

Mais de quoi parle-t-on ? il s'agit d'analyser les termes qui sont présents ensemble et d'inférer le sens des termes grâce à cette information. Le terme « voiture » est par exemple ambigu : il peut avoir le sens « automobile » ou le sens « wagon de train », il doit donc être traité pour lever l'ambigüité lorsque le moteur le rencontre.

Ainsi, lorsque l'internaute saisit « Quel est le numéro de la voiture bar dans le train ? », le moteur va analyser d'une part la probabilité d'avoir les mots « wagon », « bar » et « train » ensemble dans une requête, et d'autre part la probabilité d'avoir « automobile », « bar » et « train » ensemble. De cette façon, il va inférer que l'internaute cherche le numéro d'un wagon d'un train, qui contient le bar.

Cela paraît simple, et ça l'est, et c'est aussi très efficace. On notera cependant qu'il ne s'agit pas d'un traitement sémantique (comme ce que l'on fait avec le knowledge graph), mais encore d'une manipulation syntaxique et statistique.

Conclusion

Il n'y a pas beaucoup de choses à dire en matière de conclusion. La reformulation de requête (et donc probablement Hummingbird) est une manière d'améliorer la réponse que le moteur fournit à l'utilisateur *via* une meilleure compréhension de son besoin informationnel (la vraie demande cachée derrière la requête).

Pour les référenceurs, l'incidence est faible, sauf pour ceux qui profitent des mauvaises formulations. Pour les autres c'est plutôt intéressant, car cela légitime le travail sur le contenu, en particulier sur la création de contenu de qualité, avec des informations factuelles très pertinentes.

***Sylvain Peyronnet**, Professeur des Universités à l'Université de Caen Basse-Normandie (http://sylvain.berbiqui.org/) et **Guillaume Peyronnet**, gérant de Nalrem Médias (http://www.gpeyronnet.fr/nalrem-medias.html). Ensemble, ils font des formations et essaient de battre les loutres à la pêche à la truite.*