RankBrain, un algorithme réellement nouveau ?

Par Guillaume et Sylvain Peyronnet

Domaine :	Recherche	Référencement
Niveau :	Pour tous	Avancé

Le mois d'octobre 2015 a fait frissonner le monde du SEO avec l'annonce par Google de son algorithme nommé RankBrain, améliorant ses résultats de recherche sur de nombreuses requêtes. Un nouvel algorithme annoncé comme très puissant et essentiel dans les rouages du moteur. Cet article décortique les annonces et les travaux de recherche de Google dans plusieurs domaines comme l'intelligence artificielle, les réseaux de neurones et l'apprentissage automatique qui sont au cœur de RankBrain, pour nous en présenter les caractéristiques principales. Un nouvel algorithme ? Peut-être, mais il est certain qu'il se base sur des concepts déjà anciens...

Les nouvelles du monde SEO à la fin du mois d'octobre 2015 ont été largement saturées par une information concernant Google, au travers de l'annonce d'un « nouvel » algorithme, nom de code « RankBrain », dont la mission est d'affiner la compréhension du moteur concernant les requêtes rares demandées par les internautes utilisant le moteur. Dans cet article, après avoir fait le tour des quelques informations que l'on retrouve dans la presse à ce sujet, nous évoquerons les résultats de recherche qui sont sans doute appliqués dans cet algorithme.

Une communication floue autour de RankBrain, que faut-il retenir ?

La communication institutionnelle et journalistique de Google nous apprend que RankBrain est un algorithme « d'intelligence artificielle » et « d'apprentissage automatique » (*Machine Learning*) qui utilise une représentation des mots sous forme de vecteurs pour deviner les mots ou les phrases qui ont des sens similaires pour ensuite améliorer les SERP.

On sait ainsi que RankBrain améliore les résultats pour environ 15% des requêtes (que Google n'a jamais eu à traiter précédemment). Les lecteurs réguliers de cette lettre ne seront pas étonnés, puisque les résultats sont traditionnellement améliorés au fur et à mesure des demandes des utilisateurs, et il est donc naturel qu'un algorithme bien conçu ait un impact direct sur les requêtes jamais vues auparavant (pour celles-ci, le premier traitement hors RankBrain est rarement bon).

On nous a également affirmé que RankBrain est en production depuis presque un an, ce qui signifie que l'impact pour les référenceurs n'a pas été suffisant pour que qui que ce soit s'en rende compte avant l'annonce officiel (il en était de même pour Hummingbird, annoncé en septembre 2013 alors qu'il était en place depuis un mois et que personne ne s'en était aperçu). Enfin, l'article [1] de Bloomberg nous apprend que Rankbrain serait le troisième plus important signal pour le ranking.

Nous allons maintenant essayer de faire quelques hypothèses raisonnables sur les personnes et les algorithmes impliqués dans le mystérieux RankBrain.

Les personnes

Dans l'article [1], il est dit explicitement qu'une équipe de cinq personnes a initié le travail autour de RankBrain. C'est Greg Corrado qui le dit, et il ajoute les noms de Yonghui Wu et Thomas Strohmann. Les noms mentionnés le sont habilement car excepté Greg Corrado, les deux autres personnes sont des ingénieurs de développement, et pas des chercheurs, ce qui ne permet pas vraiment de savoir quels sont les algorithmes en jeu.

Commençons donc par Greg Corrado. Il s'agit d'un spécialiste du Machine Learning, qui a écrit de nombreux articles sur le sujet. Il a notamment récemment contribué à l'écriture d'un article de recherche [2] expliquant comment transformer une représentation textuelle en vecteurs (voilà qui rappelle quelque chose vu plus haut). Cet article a d'autres auteurs : Tomas Mikolov, Kai Chen et Jeffrey Dean. Ce dernier est un des piliers de Google, il a participé à l'écriture des premières versions de adsense/adwords, des crawlers du moteur, etc. C'est un des plus grands spécialistes du search.

Cet article a une grande importance, et chez Google personne ne s'y est trompé puisqu'un brevet a été déposé, par les mêmes personnes, sur le même sujet [3].

Tomas Mikolov est l'auteur principal de cette recherche. Il est maintenant parti chez Facebook [4], mais il mentionne sur sa page professionnelle qu'il a travaillé chez Google à la mise en place de word2vec, un outil pour transformer les mots et paragraphes en vecteurs. Chen, Dean et Corrado sont toujours chez Google, et ils ont eu un autre brevet très intéressant [5]. Dans ce brevet, on trouve un procédé qui prend en entrée des vecteurs issus du traitement de mots et de paragraphes. Ces vecteurs encodent les concepts présents dans les textes, et ils sont notés *via* une fonction de score calculée par un réseau de neurones profond (nous sommes dans le domaine du Deep Learning avec cette notion).

L'utilisation des réseaux de neurones profond pour ce type d'opération n'est pas une première chez Google. On trouve un article de recherche sur le sujet (référence [6]). Les auteurs (Sutskever, Vinyals et Le) sont tous les trois chez Google, et sont également des spécialistes du Deep Learning. Dans l'article, ils remercient pas mal de monde, dont la Google Brain team (qui est donc celle qui a créée RankBrain, au cas où vous auriez perdu le fil).

Une fois rendu à ce point, nous avons toutes les briques algorithmiques qui vont nous permettre d'extrapoler ce que fait RankBrain.

Des mots aux vecteurs

L'idée d'utiliser des vecteurs pour représenter des textes n'est pas une nouveauté, mais avec l'approche de Google que l'on retrouve notamment dans l'outil word2vec, on est sur une méthode totalement nouvelle concernant l'information directement encodée dans les vecteurs, mais aussi nouvelle concernant les techniques utilisées pour créer les vecteurs.

La méthode utilise des réseaux de neurones à deux couches pour coder le contexte des mots dans un vecteur. Qu'est ce qu'un réseau de neurones à deux couches ? C'est un ensemble de neurones « artificiels » qui sont connectés entre eux.

Un neurone au sens de l'informatique est un objet qui va réaliser une opération sur ces entrées (il va par exemple faire la somme des valeurs qu'il reçoit en entrée), puis qui va comparer le résultat de l'opération à un seuil (est-ce que la somme des entrées est plus grande que 1 254 ?). Si la comparaison est positive (la somme est effectivement plus grande que 1 254), le neurone va émettre un signal. Un réseau de neurones à deux couches est un ensemble de tels neurones qui est architecturé en deux couches reliées entre elles. Pour les lecteurs intéressés, un post de blog rempli d'informations et avec des illustrations est disponible en [9].

Un réseau de neurones peut être entrainé pour apprendre à partir d'exemples (*via* une fonction qui va modifier les valeurs des seuils et les signaux de sortie). Et c'est ainsi qu'un réseau va pouvoir créer des représentations des mots sous forme de vecteurs, en apprenant les contextes dans lesquels on trouve les mots. Plus qu'un travail sur le mot lui-même, il s'agit donc d'un travail sur le contexte.

Au final, le vecteur qui va être créé pour chaque mot sera un vecteur de grande dimension (plusieurs centaines). Chaque composante du vecteur va représenter la relation entre le mot codé par le vecteur et le mot correspondant à la composante. Cette relation est elle-même représenté par une valeur numérique (le poids de la relation entre les deux mots).

De fait, même si la technique paraît complexe, on est encore et toujours sur la même intuition : des mots qui apparaissent en relation (co-occurrence par exemple), et plus globalement qui ont des propriétés statistiques similaires, ont toutes les chances d'avoir une sémantique proche. Un contexte est donc un ensemble de propriétés statistiques autour d'un mot. L'approche de word2vec va quantifier la probabilité de retrouver certains mots à proximité d'un mot donné, et réciproquement la probabilité d'apparition d'un mot donné quand certains mots sont déjà présents dans une phrase. On voit donc que la notion est proche de la bien connue co-occurrence.

L'avantage, une fois qu'on a des vecteurs, est qu'on peut calculer la distance entre eux *via* des algorithmes robustes, comme par exemple le cosinus de salton (voir les articles [10] et

[11]). On peut ainsi déterminer que dans un corpus documentaire en langue anglaise, le mot le plus proche de « France » est « Spain ». Une autre propriété amusante donnée par les ingénieurs de Google est que les vecteurs capturent les régularités linguistiques. Ainsi, si on soustrait au vecteur de « Paris » celui de « France » et qu'on rajoute au résultat celui de « Italie », on trouve un vecteur très proche de celui de « Rome ». De la même manière le vecteur de « king » moins celui de « man » plus celui de « woman » donne presque le vecteur de « queen » (voir articles 12 et 13 sur ce point).

Ce travail qui permet de coder des mots par des vecteurs peut également être fait pour des groupes de mots. Il est ainsi possible de travailler sur des phrases, des paragraphes, voire des documents complets (l'article [7] explique comment étendre le mécanisme des mots vers les groupes de mots).

Nous allons maintenant voir ce qui est fait (probablement) par RankBrain concernant ces vecteurs.

Un apprentissage aussi profond que l'océan ?

La base théorique de ce qui suit est présenté dans l'article [6] (pour les personnes plus qu'averties). L'idée va être de ne pas utiliser un algorithme déterministe classique pour calculer la distance entre les vecteurs (comme le cosinus de Salton par exemple), mais plutôt d'utiliser un algorithme de machine Learning, et plus précisément un réseau d'apprentissage profond.

L'apprentissage profond (ou *Deep Learning*), est un domaine qui existe depuis les années 80, mais qui a réellement pris son essor autour de 2010. Cet essor est dû à quelques chercheurs (Yann LeCun – maintenant directeur du laboratoire d'intelligence artificielle de FaceBook -, Yoshua Bengio et Geoffrey Hinton).

Nous n'allons pas expliquer ici ce qu'est le deep Learning (ce serait trop compliqué). Il suffit de garder à l'esprit que c'est une large évolution des réseaux de neurones, avec de très nombreuses couches cachées, et des fonctions d'évaluation différentes d'une couche à l'autre, qui prennent notamment en compte le niveau d'abstraction des données. Ce dernier point signifie que les premières couches travaillent sur des données de bas niveau tandis les dernières sur des données de haut niveau. Si on traite une image avec du Deep Learning, cela signifie que les premières couches vont travailler sur les pixels, tandis que les dernières observeront les formes que l'on trouve dans l'image (c'est ainsi que FaceBook arrive par exemple à reconnaître des objets dans les images).

Si l'on revient à notre problème d'origine, cela veut dire qu'un algorithme d'apprentissage profond va estimer la distance, et donc la similarité, entre vecteurs. Comme les vecteurs encodent des mots, des phrases, des textes, cela veut dire que l'algorithme va être capable de calculer la similarité entre ces blocs syntaxiques. Google va donc être capable de décider si deux phrases ont le même sens (ou au moins des sens très proches). L'espoir

derrière ce type d'algorithme est qu'il apprenne la grammaire et l'orthographe de la langue, ce qu'un algorithme comme le cosinus de Salton ne peut pas faire.

Résumé et conclusion : RankBrain, comment ça marche ?

Nous allons commencer par faire le schéma opératoire de l'algorithme. Le moteur commence par calculer (en back-office) les vecteurs correspondant à toutes les requêtes qu'il connaît.

Ensuite, quand un utilisateur tape une requête :
1. Le moteur transforme la requête en un vecteur *via* un réseau de neurones à deux couches.
2. Si cette requête n'a jamais été vue, elle est rajoutée dans la base.
3. Le moteur utilise son algorithme d'apprentissage profond pour trouver les vecteurs proches de celui de l'utilisateur dans sa base de vecteurs pré-calculés.
4. La requête de l'utilisateur est enrichie avec les mots présents dans les vecteurs proches qui ne sont pas dans le vecteur de l'utilisateur.
5. Cette nouvelle requête est utilisée pour fournir des résultats qui sont, *a priori*, meilleurs.

On remarque donc que cette méthode est très similaire à ce qui a toujours été pratiqué : une expansion de requêtes. Ce qui est nouveau, c'est l'utilisation d'algorithmes issus du Machine Learning pour faire les calculs.

En regardant le point 3, on comprend pourquoi l'algorithme est surtout efficace sur les nouvelles requêtes : les anciennes ont déjà été traitées, et les SERP associées ont déjà été améliorées, il n'est donc plus utile de faire une expansion.

En conclusion, il n'y a donc rien de nouveau sous le soleil : une méthodologie ancienne, remise au goût du jour par un nouveau type d'algorithme, qui donne de meilleurs résultats.

Références

[1] http://www.bloomberg.com/news/articles/2015-10-26/google-turning-its-lucrative-web-search-over-to-ai-machines

[2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. http://arxiv.org/abs/1301.3781

[3] Computing numeric representations of words in a high-dimensional space. https://www.google.com/patents/US9037464

[4] https://research.facebook.com/researchers/643234929129233/tomas-mikolov/

[5] Scoring Concept Terms Using a Deep Network.
https://www.google.com/patents/US20140279773

[6] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).
http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[7] Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. arXiv preprint arXiv:1405.4053.
http://arxiv.org/abs/1405.4053

[8] https://code.google.com/p/word2vec/

[9] https://jcrisch.wordpress.com/2015/04/02/les-reseaux-de-neurones/

[10] Le cosinus de Salton : un classique (méconnu) des moteurs de recherche. Philippe Yonnet.
http://recherche-referencement.abondance.com/2014/02/le-cosinus-de-salton-un-classique.html

[11] http://www.peyronnet.eu/blog/modele-vectoriel-et-cosinus-de-salton/

[12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).
http://papers.nips.cc/paper/5021-di

[13] Mikolov, T., Yih, W. T., & Zweig, G. (2013, June). Linguistic Regularities in Continuous Space Word Representations. In HLT-NAACL (pp. 746-751).
http://www.aclweb.org/anthology/N13-1 - page=784

Guillaume Peyronnet *est gérant de Nalrem Médias.* **Sylvain Peyronnet** *est co-fondateur et responsable des ix-labs, un laboratoire de recherche privé. Ensemble, ils font des formations, pour en savoir plus : http://www.peyronnet.eu/blog/*