

QUAND HAL JOUE À JEOPARDY

LE 12 AVRIL 2011 DUNCAN

Dans le combat homme vs machine, IBM remporte la mise. Battre le champion du monde d'échec Gary Kasparov à l'aide d'un superordinateur ne lui a pas suffi. Sa nouvelle machine, Watson, a vaincu les champions du jeu télévisé "Jeopardy!".

En février dernier, les amateurs de La roue de la fortune, du Millionnaire et d'Une famille en or ont tremblé : Watson, le superordinateur conçu par IBM avait vaincu les deux champions américains de **Jeopardy!**. Le combat de l'homme contre la machine franchissait une nouvelle étape. Mais IBM n'en est pas à son coup d'essai. En 1997, il avait conçu l'ordinateur **Deep Blue** qui avait battu le champion d'échec Gary Kasparov.

Pourtant, curieusement, battre un champion d'échec est plus facile pour une machine que de battre un champion de Jeopardy!. Les principes de résolutions d'un problème d'échec relève de la **recherche opérationnelle** « relativement » simple. Les algorithmes utilisés, l'**élagage alpha-beta** ou l'algorithme **Negascout** peuvent en effet se résumer en quelques lignes de **pseudo-code**. Par contre, pour gagner à Jeopardy! la tâche est plus complexe.

Jeopardy! est un jeu de questions/réponses



La réponse est : Qu'est ce que Jeopardy! ?



Jeopardy! est un jeu très populaire aux Etats-Unis qui a eu son heure de gloire en France, avec l'inénarrable Philippe Risoli à la présentation. L'originalité de Jeopardy! repose sur le fait que la réponse est la question :

Par exemple, le présentateur propose :

C'est l'homme le plus classe du monde
Et le candidat doit trouver :

Qui est Georges Abitbol ?

Pour être à égalité avec les humains, il a été décidé que Watson ne pourrait se servir que des connaissances stockées dans sa base de données et ne pouvait donc effectuer de recherches sur internet.

Les algorithmes utilisés pour résoudre ce type de problème relèvent donc d'une autre branche du domaine de l'intelligence artificielle que la résolution du jeu d'échec : le *natural language processing*, qui consiste à analyser correctement le langage humain. L'un des items les plus connus de cette discipline est le fameux **test de Turing**.

Pour autant, malgré le côté sexy et impressionnant, est-ce que ce challenge d'IBM est une réelle avancée dans le domaine des intelligences artificielles ? Deux aspects sont à considérer. Pour gagner au Jeopardy!, il faut bien sûr trouver la bonne réponse mais aussi pouvoir répondre avant les autres candidats. Ces deux aspects renvoient à deux questions différentes mais imbriquées.

Quels sont les algorithmes utilisés ?

Comment ces algorithmes ont été implémentés ?

C'est l'algorithme utilisé par Watson



La réponse est : Qu'est ce que DeepQA ?



D'une manière très très grossière, répondre à une question revient à piocher des mots-clés dans ladite question, puis les mettre en relation par le biais de différents algorithmes pour extraire de la base de données à disposition les informations pertinentes.

Pour parvenir à produire un algorithme convenable, les équipes d'IBM ne sont pas parties de zéro et ont bien évidemment bénéficié de l'état de l'art. Deux outils ont tout d'abord été appliqués au challenge Jeopardy! :

- **PIQUANT**, une première solution de *natural language processing* développée quelque années auparavant par IBM

- **OpenEphyra**, une autre solution développée par l'**Université Carnegie Mellon**, en partenariat avec IBM

Toutefois, les performances de ces algorithmes ne satisfaisaient pas aux critères pour gagner. IBM a donc développé un nouvel algorithme : DeepQA

DeepQA est un pipeline modulaire



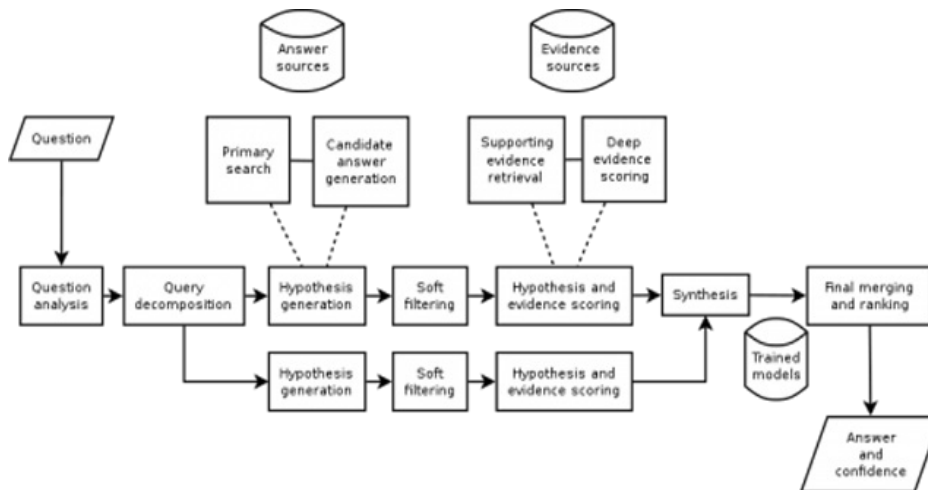
La réponse est : Quelle est l'architecture de DeepQA ?



A l'image de l'architecture de PIQUANT ou d'OpenEphyra, la clé du succès pour aboutir au résultat escompté est d'avoir conçu DeepQA comme un pipeline complet et modulaire pour y intégrer différents composants. Pour parvenir à cette architecture souple, le framework **Apache UIMA** qui intègre ces composants a été utilisé. Il avait été développé quelques années plus tôt par la même équipe.

Le processus de réponse de DeepQA peut alors se décomposer en divers blocs. A chaque étape, des dizaines d'algorithmes différents sont mobilisés. Certains provenant directement de la littérature, certains de PIQUANT ou OpenEphyra, certains ont été améliorés, d'autres

ont été imaginés par l'équipe de IBM.



En suivant le graphique ci-dessus, le processus de réponse peut être décomposé de la sorte :

Analyse de la question (*Question analysis*) : le système effectue une analyse syntaxique en extrayant des mots-clés pour déterminer comment la question sera traitée

Décomposition de la requête (*Query decomposition*) : la question est décomposée en sous-problèmes traités séparément

Génération d'hypothèses (*Hypothesis generation*) : le système propose plusieurs réponses possibles en cherchant dans sa base de données

Filtrage grossier (*Soft filtering*) : Afin de gérer au mieux les ressources de calcul, un premier tri est effectué pour éliminer les réponses les moins pertinentes

Évaluation des hypothèses et des preuves (*Hypothesis and evidence scoring*) : les réponses candidates restantes sont analysées plus en détail

Synthèse finale et classification (*Synthesis and Final merging and ranking*) : parmi toutes les réponses possibles, le système calcule la confiance accordée à chaque réponse, et classe celles-ci pour obtenir la plus probable

Réponse et confiance (*Answer and confidence*) : la réponse finale est donnée avec un niveau de confiance que le système a calculé

Tout la difficulté a été de gérer ce patchwork d'algorithmes. C'est pourquoi un gros travail a été accompli pour doter le système d'une métrique mesurant la pertinence des résultats apportées par les différentes méthodes.

Le code a été porté sur un supercalculateur



La réponse est : Comment la rapidité de Watson a été améliorée ?



La première mouture du code permettait de répondre à une question en 2 heures. Ce qui est bien insuffisant pour vaincre un compétiteur humain. Pour pouvoir répondre en moins de 3 secondes, le code a été porté sur un cluster (IBM bien sûr), regroupement de plusieurs ordinateurs appelés nœuds, travaillant de concert à la résolution du problème.

Le cluster mis à disposition est une belle bête de compétition : 90 nœuds **Power 750** contenant chacun 4 processeurs octo-coeurs de 3.5 GHz ; le tout comportant 2,880 coeurs et 16 TeraOctet de mémoire RAM pour une puissance théorique de 80 teraflops (soit 80×10^{12} ou 80 millions de millions d'opérations par seconde).

Lorsque l'on regarde le **top500** des ordinateurs les plus puissants au monde, cela le classerait aux alentours de la 110ème place. Cette puissance était d'ailleurs trop importante au regard des besoins. C'est ce qu'affirme Edward Epstein, chef du département « information non structurée » à IBM research, dont dépend le projet Watson. Durant le jeu, Watson n'a utilisé au plus que 30% des processeurs disponibles.

Porter un code sur un supercalculateur et en optimiser les performances et une tâche désormais classique dans la recherche et l'industrie. Cette étape là ne présente pas de

verrous technologiques mais demande un travail important et également beaucoup d'astuces.

Afin d'en optimiser les performances :

les données ont été mise à disposition en RAM et non sur les disques durs pour accélérer les temps d'accès

le code comportant 750,000 lignes a été parallélisé

le processus d'allocation de la mémoire, avec une **architecture NUMA**, a été optimisé

NaN



La réponse est : Est ce que c'est nouveau ?



Au vu de ce qui précède, le challenge Jeopardy! n'apporte pas d'avancées fondamentales dans le domaine théorique du *Natural Language Processing*. IBM n'a effectivement pas apporté de solution miracle au problème ou d'algorithmes révolutionnaires mais c'est appuyé sur les nombreux travaux qui existaient déjà (y compris ceux de l'entreprise elle-même).

Pour autant, il apporte bien des nouveautés. Produire une machine capable de gagner à Jeopardy! n'avait jamais été réalisé. L'intégration dans un pipeline n'avait jamais atteint ce degré d'efficacité. De plus, ce pipeline et la méthode proposée peuvent être réutilisés pour aborder d'autres domaines moins futiles.

Les limites entre ingénierie et science sont floues mais IBM apporte bien sa contribution au domaine. En fait, l'apport d'IBM est conforme au fonctionnement classique de la recherche scientifique.

C'est avant tout un très beau projet de communication



La réponse est : Que représente le le challenge Jeopardy! pour IBM ?



Sous mon clavier le terme de communication n'est pas une critique. Le travail fourni est réel et conséquent et sur bien des points du domaine, il a permis d'apporter des améliorations incrémentales indéniables. Le challenge Jeopardy! a été conçu et conduit comme un projet industriel de recherche dont l'objectif premier est de faire parler de l'entreprise. La résolution d'une véritable problématique scientifique vient en second plan.

C'est d'ailleurs **le but avoué**[pdf] de David Ferruci, le chef du projet Watson :



Roughly three years ago, IBM Research was looking for a major research challenge to rival the scientific and popular interest of Deep Blue, the computer chess-playing champion [..], that also would have clear relevance to IBM business interests.



IBM construit en effet sa stratégie de communication sur une image d'innovation radicale. Cette image permet de conquérir des marchés en faisant parler de l'entreprise. IBM se

place ainsi comme un acteur majeur du *Natural Language Processing* et promet dans un second temps des applications concrètes, comme **l'aide au diagnostic médical**. Durant ces trois ans, IBM s'est aussi rapproché des acteurs scientifiques du domaine en montant par exemple un **workshop** sur le sujet en 2008, ce qui lui a permis de rassembler autour de lui la communauté de chercheurs du domaine.

Sur le long terme, une telle stratégie permet d'attirer des talents et des collaborations avec de brillants éléments qui rêvent de travailler avec IBM. En interne, cela est source de satisfaction et de cohésion pour tous les éléments du groupe y compris ceux qui travaillent très loin de ces sujets. Travailler dans une entreprise ayant une bonne image est valorisant. C'est également un élément de motivation et de promotion interne. Il permet de faire miroiter à des ingénieurs, qui ont une tâche routinière au quotidien, l'opportunité de pouvoir un jour rejoindre ce type de projets au sein du groupe.

Dans ce cas là, ne pourrait-on pas dire que Watson est un projet qui permet à IBM de gagner sur de nombreux tableaux ? Tout comme à Jeopardy!, la réponse est dans la question.

POUR ALLER PLUS LOIN :

Building Watson: An Overview of the DeepQA Project [pdf], AI Magazine Vol 31, N°3, p. 59-79, l'article principal utilisé pour ce billet. Rédigé par l'équipe en charge de Watson, il explique leur démarche et le fonctionnement détaillé de deepQA.

La page d'accueil d'IBM research sur ses activités de natural language processing
Le **blog** d'IBM research

La page Quora sur le Watson Jeopardy Challenge

La page wikipedia (eng) sur Watson

Watson Takes a Turn on Wall Street, résumé de la rencontre entre Edwar Epstein et Michael Feldman, éditeur de HPCwire.

Elémentaire, mon cher Watson, le billet de de Choux Romanesco

>> Photo Flickr CC-BY-NC-SA  par **jntolva**

>> Article initialement publié sur **Nanostelia**