Intelligence Artificielle et langage

BERNARD VICTORRI

UN SURVOL HISTORIQUE RAPIDE

Le langage a été un enjeu crucial pour l'IA dès ses débuts au milieu du siècle dernier. En témoigne l'article fondateur d'Alan Turing (1950), qui, pour répondre à la question *Can machines think?* (Les machines peuvent-elles penser ?), propose le fameux jeu de l'imitation : une machine et un humain dialoguent par écrit avec un expérimentateur, et si celui-ci ne peut détecter lequel de ses interlocuteurs est une machine, alors la machine sera dite « intelligente ». Nul doute que les systèmes d'IA générative, apparus ces dernières années, passeraient haut la main le test de Turing. Il aura donc fallu quelques 70 ans, bien des tâtonnements, et d'énormes progrès technologiques, pour réaliser entièrement le programme de recherche inspiré par Turing.

Dès les premiers développements, deux approches concurrentes se sont dégagées. La première, que l'on a appelée *l'approche symbolique*, a cherché à modéliser la pensée humaine, en donnant de manière explicite à la machine les règles de fonctionnement du raisonnement humain : cela a conduit notamment dans les années 70-80 au développement des systèmes experts qui ont suscité un véritable engouement, mais l'enthousiasme est retombé par la suite quand sont apparues les difficultés de rendre compte par ces méthodes de la complexité des processus à l'œuvre chez les experts humains, pour lesquels l'intuition et l'expérience acquise inconsciemment jouent un rôle au moins aussi important que le raisonnement logique conscient. La deuxième approche, que l'on a appelée *l'approche connexionniste*, a consisté à modéliser directement l'activité du cerveau humain, à l'aide de réseaux de neurones formels imitant (en le simplifiant) le fonctionnement des neurones biologiques, les poids des interactions entre neurones étant modifiés par apprentissage jusqu'à l'obtention d'un taux de réussite convenable pour une tâche donnée. Cette approche a connu ses premiers vrais succès dans les années 80, pour des tâches de reconnaissance de forme, après que soit mise au point une méthode très efficace d'apprentissage (basée sur *l'algorithme de rétropropagation du gradient*, cf. Le Cun 1985).

En ce qui concerne le traitement du langage, c'est l'approche symbolique qui a dominé durant ces années. Les recherches se sont focalisées sur la manière de représenter le sens, ce qui impliquait de trouver un moyen de structurer les connaissances humaines à l'œuvre lors de la compréhension d'un texte (cf. par exemple les graphes conceptuels de Schank 1969, ou encore la théorie des *frames* de Minsky 1974). Dans l'approche symbolique, les théories linguistiques sont fortement sollicitées, notamment pour réaliser l'analyse syntaxique des textes en réception et en production. Mais malgré tous ces efforts, les résultats sont peu convaincants. Le sens des phrases et des textes se révèle très difficile à formaliser, en dehors de quelques domaines fermés très restreints. Quant à l'approche connexionniste, elle semble encore moins efficace, malgré la mise au point d'un type de réseaux de neurone bien adaptés au traitement de données séquentielles, *les réseaux récurrents* (Elman 1990).

Au cours des années suivantes (1990-2005) apparaissent de nouvelles approches, à mi-chemin entre les approches symbolique et connexionniste, caractérisées par une utilisation de plus en plus grande de méthodes quantitatives fondées sur l'analyse statistique de volumes très

importants de données (*big data*). Cette évolution est favorisée par les progrès de la technologie, permettant de traitements massifs qui étaient impensables auparavant. Pour nous en tenir au traitement du langage, c'est à cette époque que la traduction automatique prend tout son essor, avec l'exploitation de grands corpus bilingues : on renonce à essayer de représenter le sens des textes que l'on veut traduire, mais pour chaque mot ou groupe de mots (syntagme), on prend en compte le contexte dans lequel il est inséré pour obtenir une traduction correcte, grâce aux exemples proches que l'on extrait des corpus bilingues dont on dispose (cf. Koehn 2003). Il semble donc acquis à cette époque que les deux approches traditionnelles ont en partie échoué : pour réaliser les objectifs de l'IA, plutôt que de chercher à imiter la pensée ou le cerveau humain, mieux vaut s'attaquer à chaque problème indépendamment de manière pragmatique, sans a priori théorique, en utilisant toute la puissance de calcul et les masses de données offertes par les avancées technologiques.

Mais les choses vont radicalement changer à partir de 2007. En effet, cette année là trois spécialistes des réseaux de neurones, Yann Le Cun, Geoffrey Hinton et Yoshua Bengio, présentent des résultats inégalés avec des réseaux de très grande taille : c'est l'acte de naissance de *l'apprentissage profond* (*deep learning*), qui est d'abord appliqué à la reconnaissance de formes (cf. Bengio & LeCun 2007). L'architecture des réseaux utilisés rappelle par bien des aspects celle du système visuel des mammifères : un grand nombre de couches successives, analysant les images d'un niveau très local au niveau global, grâce à des opérations de même type (des *convolutions*) appliquées à différentes échelles. Les performances sont impressionnantes : pour la première fois, elles égalent les performances humaines sur des tâches spécifiques, à commencer par la reconnaissance de chiffres manuscrits.

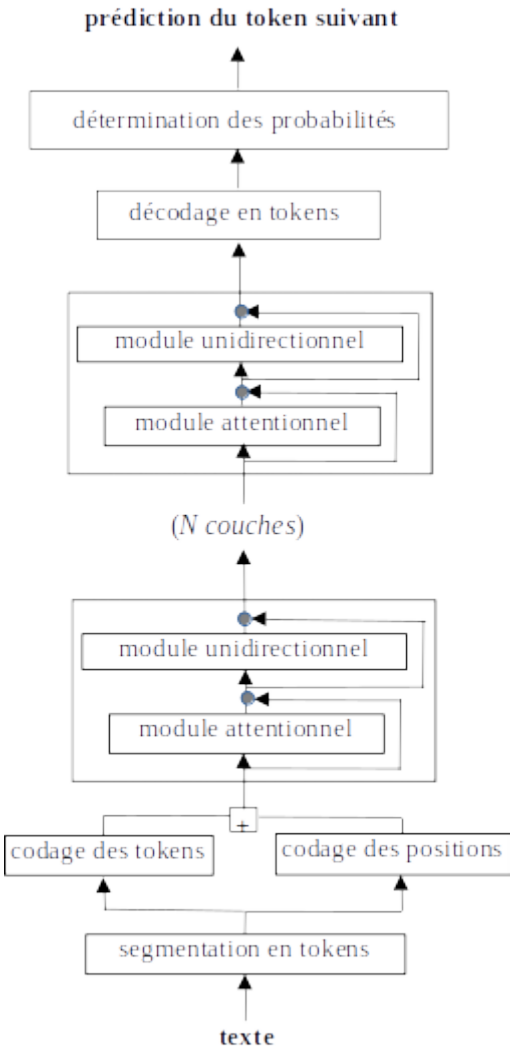
Les techniques de l'apprentissage profond se sont alors très vite répandues dans tous les domaines de l'IA. Une étape remarquable a été franchie avec la maîtrise totale du jeu de go, réputée impossible pour une machine ; à noter que le meilleur programme fonctionne sans aucune connaissance au départ hormis les règles du jeu, ni aucun exemple de partie jouée par des humains, mais uniquement par de l'auto-apprentissage en jouant contre lui-même (Sylver *et al.* 2017).

Dans le domaine du langage, l'apprentissage profond a aussi opéré une révolution, d'abord avec des architectures assez complexes, à base de modules récurrents, puis, à partir de 2017, avec une architecture beaucoup plus simple mais des tailles de réseau toujours plus gigantesques, à la suite d'un article fondateur d'une équipe de recherche de Google (*Attention is all you need*, Vaswani A. *et al.* 2017). C'est le début des système d'*IA générative*, qui vont très vite s'imposer par leurs performances spectaculaires. Comme on le verra à la section suivante, ces systèmes, dont la taille commence à se rapprocher de celle du cerveau humain, réalisent sans aucun doute l'objectif de l'approche connexionniste de l'IA : grâce à leur apprentissage intensif, ils sont capables de s'adapter à différentes tâches langagières ; tout se passe comme s'ils avaient acquis la capacité de comprendre les demandes de leur interlocuteur et d'y répondre de façon pertinente, et donc comme s'ils maîtrisaient le sens des textes auxquels ils sont soumis. Mais nous n'avons pas accès, du moins pas encore, à la manière dont s'organise ces représentations internes du sens, comme cela est d'ailleurs aussi le cas pour le cerveau humain...

Les systèmes d'IA générative

Architecture

Les systèmes d'IA générative utilisent un réseau de neurones appelé *transformeur* (Radford *et al.* 2018 ; Brown *et al.* 2020). Un transformeur est un réseau composé d'un grand nombre de couches (de l'ordre de la centaine, voire plus…), chaque couche étant elle-même composée de deux modules, un module attentionnel et un module unidirectionnel. À cela s'ajoute un module de codage en entrée et un module de décodage en sortie du réseau (cf. figure ci-contre). L'entrée est un texte qui va être segmenté en *tokens* (ce sont pour l'essentiel des mots, mais pas uniquement). L'ensemble des tokens utilisables constitue donc le dictionnaire du système ; sa taille est de l'ordre de plusieurs dizaines de milliers de tokens. La sortie est une distribution de probabilités sur l'ensemble des tokens du dictionnaire.



Sans entrer dans les détails ici, disons que le codage consiste à représenter chaque token du texte par un vecteur de très grande dimension (de l'ordre de plusieurs milliers). C'est cette suite de vecteurs qui est présenté aux couches du réseau, qui vont les transformer progressivement. Le décodage consiste à repasser des vecteurs transformés à des tokens : chaque token du dictionnaire reçoit une valeur qui va fournir la probabilité que ce token soit choisi en sortie.

Comme dans tous les réseaux de neurones, les transformations (y compris le codage et le décodage) sont déterminées par les poids des connexions entre neurones. Ces poids, qu'on appelle les *paramètres* du réseau, sont obtenus par apprentissage, comme on va le voir ci-dessous. Le nombre de paramètres de ces réseaux est énorme : de l'ordre du billion (par comparaison, le cerveau humain comporte seulement mille fois plus de connexions…).

La grande innovation des transformeurs, c'est la présence des modules attentionnels dans les couches intermédiaires. Ces modules permettent de tenir compte des relations à distance entre différents mots du texte, par exemple entre un verbe et son sujet, ou entre un pronom et le nom auquel il réfère. Ils sont donc très puissants, mais réclament beaucoup de connexions. Les modules unidirectionnels sont beaucoup plus classiques dans les réseaux de neurones : ils transforment chaque vecteur de la suite indépendamment, et dans un module donné les différents vecteurs subissent la même transformation. À noter enfin que chaque module est muni d'un "court-circuit" qui consiste à recombiner l'entrée et la sortie du module : c'est ce mécanisme qui permet d'avoir un très grand nombre de couches sans que le signal d'entrée ne s'affaiblisse trop en cours de route.

Fonctionnement

Le fonctionnement du réseau est simple : on lui soumet un texte, que l'on appelle une *invite* (*prompt*). On obtient en sortie un token, choisi suivant la distribution de probabilité obtenue en sortie du réseau. On ajoute ce token en queue de l'invite et on soumet le texte ainsi augmenté : on obtient un second token, et l'on recommence autant que l'on veut ce processus, générant ainsi tout un texte qui prolonge le texte initial. Si la taille maximale de l'entrée est atteinte (elle est de plusieurs milliers de tokens au moins, tout de même), on supprime un token en tête du texte chaque fois que l'on rajoute en queue un nouveau token.

Pour que le texte ainsi produit ait du sens, il faut que le réseau ait appris à prédire correctement quels tokens sont susceptibles de suivre le texte présenté en entrée. Pour cela, il faut passer par une phase d'apprentissage, longue et coûteuse (plusieurs dizaines de jours, sur les plus gros ordinateurs actuels, avec des processeurs spécialisés et beaucoup de parallélisme…), en utilisant un corpus de textes tout-venant multilingue lui aussi de taille impressionnante (des milliers de milliards de tokens). L'apprentissage consiste à prélever au hasard un bout de texte dans cet immense corpus, à le présenter comme invite, à comparer le token produit en sortie avec le token qui suivait le bout de texte dans le corpus, et à corriger les poids du réseau en conséquence. Cette correction est donnée par l'algorithme de rétropropagation du gradient, un algorithme très efficace qui est à l'origine des succès de l'approche neuronale depuis le début de l'IA. On réitère la procédure jusqu'à l'obtention d'un taux de succès satisfaisant.

Après cette phase d'apprentissage, on constate que le système produit des textes de bonne qualité sur le plan syntaxique, sémantique et discursif : autrement dit, ces textes sont bien écrits, cohérents et pertinents par rapport au texte initial. Et cela pour une grande variété de langues (sans parler de ses compétences dans une dizaine de langages informatiques). Ce résultat est déjà en soi assez extraordinaire : il semblait complètement inaccessible il y a à peine dix ans. Cependant il faut encore une dernière phase de traitements pour que le système soit vraiment opérationnel et qu'il réponde aux demandes précises des utilisateurs. En effet, celui-ci a généralement un besoin spécifique, qui peut être de nature très diverse : résumer un texte, écrire une nouvelle, rédiger un rapport, écrire un programme informatique, traduire un texte, répondre à une question, commenter une information, ou encore, comme dans notre cas, classer des textes par thématique… Or, comme on l'a vu, le système ne sait faire qu'une chose à l'issue de son apprentissage : prolonger le texte qu'on lui a soumis de manière pertinente, mais pas forcément dans le sens attendu par l'utilisateur.

Pour surmonter cette difficulté, il faut une nouvelle phase d'apprentissage, beaucoup plus légère celle-là, qu'on appelle le *peaufinage* (*fine-tuning*, cf. Ouyang et al. 2022). Cela consiste à noter les différentes réponses du système à une invite donnée suivant qu'elles satisfont plus ou moins bien les attentes de l'utilisateur, et à modifier légèrement les paramètres de façon à privilégier les réponses adaptées à sa demande. Après cette dernière étape, le système donne vraiment l'impression de comprendre ce que lui dit son interlocuteur humain, quel que soit le type de demande qu'on lui adresse, comme tout un chacun peut s'en convaincre en utilisant les systèmes grand public qui ont proliférer au cours de l'année 2023 : *ChatGPT* d'OpenAI, *Copilot* de Microsoft, *Gemini* de Google, *LlaMA* de Meta, ou encore *Le Chat* de l'entreprise française Mistral

(pour une analyse approfondie des performances de GPT-4, le plus récent système d'OpenAI, voir Bubeck 2023).

Il faut souligner que la réussite de la dernière étape, le peaufinage, n'est possible que parce que l'apprentissage initial, centré sur la tâche de prédiction du token suivant, a eu comme effet de bord, au-delà de la tâche demandée, de faire du réseau une véritable machine de compréhension du langage : tout se passe comme si le système avait acquis la capacité de représenter le sens d'un texte à l'intérieur du réseau, de la même manière que nos neurones sont effectivement capables de telles représentations internes, ce que nous traduisons en disant que nous avons compris ce que nous lisons ou entendons.

Cela ne veut pas dire que ces systèmes soient sans défaut. Notamment ils peuvent très bien produire des informations fausses, voire inventer complètement des événements qu'ils présentent comme s'étant réellement déroulés : c'est ce que l'on appelle le phénomène d'*hallucination*. Il est très difficile de corriger ce défaut, qui est structurellement lié à la manière dont ces systèmes ont été conçus. En effet, au moment où le système répond à un utilisateur, il n'a plus accès au corpus qui a servi à son apprentissage, il est donc dans une situation analogue (toute proportion gardée) à celle un étudiant qui aurait essayé d'apprendre par cœur une encyclopédie et à qui l'on demanderait lors d'un examen de disserter sur tel phénomène physique ou telle période historique : il est clair que l'on aurait pas intérêt à ne pas prendre pour argent comptant l'intégralité de ses réponses ! D'une certaine manière on pourrait en conclure qu'en réussissant à modéliser (en partie) le fonctionnement du cerveau, on a développé des systèmes qui ont aussi hérité de ses imperfections…

Bibliographie

Bengio Y. & Le Cun Y. (2007). Scaling learning algorithms towards AI. Bottou L., Chapelle O., DeCoste D. & Weston J. (éds), *Large-Scale Kernel Machines*, MIT Press, 321-359.

Bubeck S. *et al.* (2023). Sparks of Artificial General Intelligence: Early experiments with GPT-4, *arXiv:2303.12712v3.*

Brown T. *et al. (*2020). Language Models are Few-Shot Learners, *arXiv:2005.14165v1.*

Elman J.L (1990). Finding Structure in Time. *Cognitive Science*, *14*(2), 179-211.

Koehn P., Och F.J. , Marcu D. (2003). Statistical phrase based translation. *Proceedings of the Conference on Human Language Technologies*, 127-133.

LeCun Y. (1985). Une procédure d'apprentissage pour réseau à seuil asymétrique. *Proceedings of Cognitiva 85*, 599-604, Paris.

Minsky M. (1974). A Framework for Representing Knowledge. *MIT-AI Laboratory Memo 306*, MIT, Boston, Mass.

Ouyang L. et al. (2022). Training language models to follow instructions with human feedback, arXiv:2203.02155v1.

Radford A. *et al. (*2018). Improving Language Understanding by Generative Pre-Training, *OpenAI report.*

Schank, Roger (1969). A conceptual dependency parser for natural language. *Proceedings of the 1969 conference on Computational linguistics.* Sång-Säby, Sweden. 1–3.

Silver D., Schrittwieser J. , Simonyan K. *et al.* (2017). Mastering the game of Go without human knowledge, *Nature*, 550 (7676), 354–359 .

Turing A. M. (1950). Computing machinery and intelligence. *Mind*, 49, 433-460.

Vaswani A. *et al.* (2017). Attention Is All You Need, *arXiv:1706.03762v5.*