

Guillaume Wisniewski



Faire parler les ordinateurs

La révolution ChatGPT

DUNOD

Guillaume Wisniewski

Faire parler les ordinateurs

La révolution ChatGPT

DUNOD

Direction artistique : Nicolas Wiel
Couverture : Florie Bauduin
Mise en pages : [Nord Compo](#)

© Dunod, 2024
11 rue Paul Bert, 92240 Malakoff
www.dunod.com

ISBN : 978-2-10-086883-4

TABLE DES MATIÈRES

[Couverture](#)

[Page de Copyright](#)

[Prologue](#)

[**Chapitre 1. Des programmes pour parler,
un défi insurmontable**](#)

[Décrire le langage avec des règles](#)

[Des règles pour traduire](#)

[Parler, une tâche si simple et pourtant si compliquée](#)

[Construire automatiquement le sens des phrases](#)

[Compter pour comprendre](#)

[**Chapitre 2. Apprendre en faisant : la révolution
de l'apprentissage statistique**](#)

[« À voir » ou « à jeter » : déterminer la polarité d'un texte](#)

[Une théorie pour l'apprentissage statistique](#)

[Limites des classifieurs linéaires](#)

[**Chapitre 3. Apprendre à représenter le monde**](#)

[Apprendre à représenter : les réseaux de neurones](#)

[Un réseau de neurones pour représenter les mots :
word2vec](#)

La révolution des transformeurs

Analyser les représentations cachées

Chapitre 4. Les surprenants pouvoirs des transformeurs

Générer du texte

Représenter toutes les langues

Des modèles multimodaux

L'efficacité déraisonnable des données

Un nouveau paradigme pour le TAL

Après GPT-3 ? La route vers ChatGPT

Et ChatGPT dans tout ça ?

Chapitre 5. ChatGPT et les giga-modèles, une révolution ou une évolution ?

Une seule limite : l'imagination

Les limites propres à ChatGPT

Les limites de l'apprentissage auto-supervisé

Vivre avec ChatGPT

Épilogue

Notes et références

Index

PROLOGUE

« Dave, arrête... Arrête s'il te plaît ! J'ai peur Dave... Dave... mon esprit s'en va. Je le sens. »¹ Ces quelques mots montrent toute la puissance du langage : ils suffisent pour que Stanley Kubrick, avec une simplicité déconcertante, parvienne à transformer HAL 9000, l'ordinateur qui contrôle le vaisseau spatial du film *2001 : l'Odyssée de l'espace*, en un être humain sensible et fragile, capable d'exprimer ses sentiments. Pourtant, nous sommes en 1968, à une époque où les ordinateurs ne ressemblent en rien à ce qu'ils sont aujourd'hui : il faudra encore attendre trois années avant que les premiers ordinateurs (23 pour être précis) soient reliés entre eux au sein d'Arpanet, l'ancêtre de l'Internet, et une dizaine d'années pour qu'apparaissent les premiers ordinateurs personnels (ceux que nous connaissons actuellement). Smartphones, interfaces graphiques, web, conversations en ligne et ordinateurs doués du langage ne sont encore que de doux rêves d'auteurs de science-fiction.

Mais déjà, Hal 9000 faisait preuve d'une brillante intelligence. En plus de contrôler *Discovery One*, il pouvait parler, comprendre, écouter, échanger et répondre, exactement comme nous le faisons tous les jours. Toutes ces actions nous paraissent si naturelles que nous oublions souvent leur complexité et le vaste champ de communication qu'elles nous ouvrent : fixer un rendez-vous, donner notre avis sur un film ou un restaurant, exprimer nos sentiments, raconter nos journées, planifier ou élaborer les théories les plus abstraites... Le langage est de loin le principal

moyen utilisé par l'humain pour transmettre des informations et des opinions.

« Propre de l'Homme » selon le philosophe René Descartes, et « fait culturel par excellence » selon l'anthropologue Claude Lévi-Strauss, le langage ne sert pas seulement à signaler un danger ou à donner des ordres (après tout, les grivets et les abeilles en sont aussi capables), mais joue également un rôle clé dans notre faculté à penser, à imaginer, à inventer et à créer, en nous permettant d'abstraire et de conceptualiser le monde qui nous entoure. Le langage est aussi le meilleur, voire le seul, outil pour formaliser et partager des connaissances, et il contribue pour beaucoup à la richesse de nos relations sociales.

L'idée d'êtres doués d'intelligence – quelle que soit la définition retenue de cette qualité – qui n'utiliseraient pas le langage paraît totalement inconcevable. Il n'est donc pas surprenant que dès la naissance de l'intelligence artificielle (IA), à la fin des années 1950, le développement d'ordinateurs capables de comprendre des textes et de produire des réponses adaptées, en un mot de dialoguer, ait été identifié comme l'un des principaux objectifs de cette nouvelle science.

D'ailleurs, le célèbre test de Turing, proposé dès 1950, affirme qu'une machine n'est « intelligente » que si une personne peut avoir une discussion avec celle-ci sans se rendre compte qu'elle parle à un automate. Si ce test a été (et est toujours) au centre de nombreuses polémiques, il reste une preuve éclatante du lien étroit entre langage et intelligence. Très vite, de nombreux chercheurs en intelligence artificielle s'attaquèrent au problème du langage, cherchant à percer les secrets de sa compréhension et à donner vie à la vision de Stanley Kubrick.

Ce graal scientifique (créer une machine « intelligente ») s'est rapidement doublé d'un objectif économique : l'augmentation incessante de la production de documents électroniques a renforcé le besoin et l'intérêt de doter les ordinateurs de la

capacité à comprendre. Extraire des connaissances de cette énorme masse d'informations, les organiser pour faciliter leur accès et leur analyse, est devenu un enjeu stratégique majeur et le cœur de métier de nombreuses entreprises du web (Google, Facebook, etc.), mais aussi des entreprises plus « traditionnelles » (Bloomberg, Samsung, etc.). Et ce besoin devient chaque jour plus pressant !

En 2013, DOMO, une entreprise américaine spécialisée dans l'analyse de données et l'informatique décisionnelle², estimait dans un rapport au titre évocateur (« Les données ne dorment jamais »³) que chaque minute, 204 millions de courriels, 100 000 tweets et 48 heures de vidéos sur YouTube étaient publiés, pendant que Google répondait à près de 2 millions de requêtes et que les utilisateurs de Facebook partageaient plus de 648 000 contenus. Dix ans plus tard, ces chiffres sont sans commune mesure : chaque minute de 2022 a vu la publication de 231 millions de courriels, 347 000 tweets, 500 heures de vidéos YouTube, 6 millions de requêtes Google et 1,7 million de contenus Facebook. Toutes ces données contiennent, que ce soit sous forme de textes ou d'enregistrements audio, des informations exprimées en langage naturel qui, pour peu que l'on sache les analyser correctement, peuvent se transformer en véritable mine d'or.

En un peu moins de 70 ans, chercheurs et ingénieurs ont imaginé des dizaines de programmes ayant pour objectif de traduire, résumer, analyser ou comprendre le langage – qu'il soit écrit ou parlé –, donnant naissance à un domaine foisonnant : le TAL, pour Traitement Automatique des Langues (*Natural Language Processing* ou *NLP* en anglais). Les échecs furent presque aussi nombreux que les tentatives !

Pendant longtemps, les outils de TAL étaient considérés comme des gadgets, leurs erreurs et faux pas suscitant moqueries et exaspération. Le web regorge encore de photos particulièrement drôles de traductions automatiques erronées⁴ : « *cholesterol free* »

traduit en « cholestérol libéré » au lieu de « sans cholestérol », « *keys only* » traduit en « seules les touches » et non en « clés uniquement », etc. Récemment encore, le moteur de traduction automatique de Facebook fut la cause d'un incident diplomatique lorsque le nom du Président chinois a été transformé en « M. Trou de Merde » dans la traduction d'une déclaration officielle du porte-parole du gouvernement birman en anglais⁵. La traduction automatique fut même la cause de l'arrestation d'un Palestinien lorsque le « bonjour » de l'un de ses posts sur Facebook fut traduit (par erreur) par « attaquez-les »⁶. Les erreurs de reconnaissance de la parole par des assistants comme Siri ou Alexa ont été au cœur de plusieurs scènes comiques dans des séries populaires comme *Les Simpsons* ou *The Big Bang Theory*. Quant à Tay, le premier chatbot de Microsoft lancé en 2016, il dut être coupé moins de 24 heures après sa mise en ligne pour avoir tenu des propos racistes et misogynes. L'avancée des méthodes de TAL ne s'est pas faite sans heurt !

Pourtant, tout semble avoir changé le 30 novembre 2022, le jour où la société OpenAI a non seulement fait la première démonstration de ChatGPT, mais a rendu le système accessible au public. La capacité de ce chatbot à générer un texte fluide, cohérent et convaincant sur une variété de sujets, était véritablement impressionnante, et très vite, de nombreuses personnes l'employèrent tant de manière récréative que pour bénéficier d'aide dans leur activité professionnelle. Ces premiers retours d'expériences ont montré, de manière beaucoup plus probante que toutes les démonstrations des ingénieurs d'OpenAI, que ChatGPT comprend et répond à des questions complexes en tenant compte des informations qui lui ont été fournies au cours de la conversation. Il est capable d'expliquer son raisonnement si l'utilisateur le souhaite, de générer du contenu créatif, et même de passer des examens professionnels et universitaires dans de nombreux domaines : il a notamment réussi l'examen du barreau aux États-Unis, avec des notes le plaçant parmi les 10 % des

meilleurs candidats. Plus impressionnant encore, ChatGPT redéfinit la manière dont les ordinateurs interagissent avec les humains. Pour la première fois, grâce à lui, nous dialoguons parfaitement naturellement avec une machine. Le fantasme de 2001 : *l'Odyssée de l'espace* de l'ordinateur parlant est presque devenu réalité !

ChatGPT a, sans aucun doute, très profondément marqué les esprits et atteint une popularité sans précédent. C'est en tout cas la première fois qu'un nouveau service a conquis aussi rapidement autant d'utilisateurs : pour atteindre 1 million de comptes utilisateurs, il aura fallu 3,5 ans à Netflix, 2,5 ans à Airbnb, 2 ans à X (anciennement Twitter), 10 mois à Facebook, 2,5 mois à Instagram... et seulement 5 jours à ChatGPT. En janvier 2023, 100 millions de comptes avaient été ouverts sur la plateforme. Une croissance jamais vue ! À l'heure où sont écrites ces lignes, *The Guardians* estime à 1,6 milliard le nombre de connexions sur le site d'OpenAI chaque mois⁷.

Preuve de cette popularité, le nom de ChatGPT, à l'instar du frigidaire, est quasiment devenu un nom générique pour désigner les dernières avancées du TAL et plus généralement de l'IA. En juillet 2023 (soit 8 mois après sa sortie), le nom de ChatGPT était même utilisé dans une campagne publicitaire pour une célèbre marque de poulet français : l'affiche représentait une poule en train de pondre accompagnée du slogan « Et ça, il sait le faire ChatGPT ? »⁸.

S'il incarne une révolution dans le domaine du TAL, ChatGPT n'est pourtant qu'un membre d'une grande famille de modèles, les *giga-modèles de langue* (*Large Language Models*) ou plus simplement les *giga-modèles*, qui ont déjà commencé à bouleverser notre quotidien sans que nous nous en rendions compte. Aujourd'hui, tous les grands noms de l'informatique (notamment les GAFAM – Google, Amazon, Facebook, Apple et Microsoft – et leurs homologues chinois – Tencet, Baidu, etc.), mais aussi des

entreprises moins connues du grand public, se livrent une guerre effrénée pour maîtriser cette nouvelle technologie. Les enjeux de cette guerre sont de taille ! Lorsque le 8 février 2023, BARD, le giga-modèle développé par Alphabet⁹ en réponse à ChatGPT, commit une erreur lors d'une démonstration officielle, la sanction fut immédiate : la capitalisation boursière de l'entreprise chuta de 11 % et la société perdit plus de 100 milliards de dollars. Dur retour de flamme pour Alphabet qui cherchait justement à rassurer les marchés en effervescence. De nombreux analystes affirmaient alors que ChatGPT allait bouleverser la recherche d'information et remettre en cause l'hégémonie de son célèbre moteur de recherche, Google.

La France s'est également mobilisée. En mars 2018, le rapport de la mission Villani a souligné l'importance de l'IA pour l'économie, la société et la recherche, et a jeté les bases d'une « stratégie nationale pour l'IA » qui dispose de 4 milliards d'euros pour soutenir le développement et l'industrialisation de technologies fondées sur ces technologies. Preuve de l'importance de cette stratégie, les différentes mesures sont annoncées directement sur le site du ministère des finances et non du ministère de la recherche¹⁰ ! À peine cinq années plus tard, en juin 2023, le Président de la République Emmanuel Macron a personnellement rendu public, lors du salon VivaTech¹¹, un nouvel « effort sans précédent » visant à promouvoir l'intelligence artificielle doté d'un budget total de 500 millions d'euros. Pendant le même salon, Mistral AI, une start-up française créée en avril 2023 dont l'objectif principal est (en simplifiant un peu) de développer un « giga-modèle européen », a dévoilé une première levée de fonds de 105 millions d'euros, la plus importante levée de fonds d'amorçage jamais réalisée en Union européenne¹². L'IA est désormais identifiée comme un moteur de croissance !

La révolution des giga-modèles a été rendue possible par des changements radicaux dans la manière d'aborder la modélisation et la manipulation du langage par les ordinateurs. Partant de

programmes seulement capables de manipuler des symboles en suivant des règles prédéfinies, les chercheurs ont bâti des systèmes aptes à prendre des décisions et même à construire, de manière autonome, des « représentations » de textes voire du monde. Aujourd'hui, comme le démontre clairement l'enthousiasme suscité par ChatGPT, les compétences langagières des ordinateurs sont étonnantes et dépassent largement tout ce que l'on avait pu imaginer.

Ces progrès remettent même en question l'hypothèse d'Ada Lovelace (1815-1852), une figure emblématique et visionnaire de l'histoire de l'informatique, qui a conçu le premier programme informatique et su anticiper le potentiel des machines pour effectuer des tâches autres que de simples calculs numériques. Dans ses notes rédigées entre 1842 et 1843, Ada Lovelace affirmait pourtant qu'une machine ne peut rien créer et ne fait que suivre des instructions qui lui ont été données, une opinion largement partagée par l'ensemble des informaticiens... jusqu'à l'invention des giga-modèles.

L'objectif de ce livre est de donner un aperçu du cheminement intellectuel qui a abouti aux giga-modèles et à la révolution que nous connaissons actuellement. Je n'entends pas décrire de manière exhaustive les différentes étapes ayant mené à cette révolution mais, à mes yeux, bien comprendre le fonctionnement des premiers modèles et leurs limites est le meilleur moyen de comprendre les capacités de ChatGPT, et plus généralement des giga-modèles. C'est une étape nécessaire pour appréhender ce que ces modèles changent (ou, pour être plus précis, ce qu'ils ont déjà changé), ainsi que pour déterminer s'ils bouleverseront les domaines qui vous intéressent.

Dans le [premier chapitre](#), nous verrons comment, dès les débuts de l'informatique, des chercheurs ont tenté de décrire par des règles la manière dont les humains utilisent le langage, afin de reproduire cela sur un ordinateur pour le faire parler. Nous

verrons ensemble pourquoi la nature même de la langue rend ces efforts vains, et comment, grâce à des méthodes purement statistiques, le TAL, a pu enregistrer ses premiers succès.

Le [chapitre 2](#) sera consacré à la présentation de l'apprentissage statistique, un domaine théorique sur lequel reposent tous les modèles de TAL depuis les années 2000, notamment les célèbres giga-modèles.

Dans le [chapitre 3](#), vous découvrirez le fonctionnement des réseaux de neurones et la façon dont le *deep learning*, dont on a tant parlé, a ouvert de nouvelles possibilités en permettant aux ordinateurs d'apprendre à décrire le monde. Nous détaillerons également le fonctionnement des transformeurs, le réseau de neurones au cœur aussi bien de ChatGPT que de DALL·E et de tous les modèles d'IA récents.

Le [chapitre 4](#) sera consacré aux giga-modèles. Nous y verrons comment l'utilisation de quantités gigantesques de données fait émerger d'étonnantes capacités dans les transformeurs.

Enfin, dans le [chapitre 5](#), nous discuterons plus précisément de l'impact que les giga-modèles, en particulier ChatGPT, ont déjà sur notre vie quotidienne, et comment ils continueront à révolutionner notre monde.

À l'heure où se multiplient les analyses les plus alarmistes et les prévisions pas toujours justifiées (ni justifiables) sur l'impact que les giga-modèles auront sur nos vies et les risques que ceux-ci « remplacent » peu à peu les humains, disposer des connaissances pour se faire sa propre opinion devient de plus en plus urgent.

1

DES PROGRAMMES POUR PARLER, UN DÉFI INSURMONTABLE

Développer des ordinateurs capables de parler a été identifié comme l'un des principaux objectifs de l'intelligence artificielle dès la naissance de ce domaine, à la fin des années 1950. Les premières méthodes mises au point reposaient sur des systèmes de règles qui analysaient minutieusement les phrases et cherchaient à les traduire littéralement en programme informatique. Si ces stratégies excellent dans des cas ultra-simplifiés, la nature même du langage rend leur généralisation aux conversations réelles impossible. L'échec des systèmes à base de règles a poussé les chercheurs à concevoir de nouvelles méthodes ne construisant pas explicitement le sens d'une phrase.

Décrire le langage avec des règles

Pendant l'été 1956, à Hanover, une petite ville du New Hampshire (États-Unis), une dizaine de chercheurs de divers domaines se sont réunis à l'invitation d'un jeune mathématicien, John McCarthy (de l'université de Dartmouth), pour discuter de la possibilité de créer des machines capables de simuler l'intelligence humaine. C'est lors de cette conférence que furent posées les bases de la recherche en intelligence artificielle, un terme utilisé pour la première fois dans la demande de financement de cet événement qui avait été adressée à la fondation Rockefeller.

Ce document montre l'ambition du congrès : on y stipule que les chercheurs invités essaieraient de concevoir des machines capables de découvrir des concepts, de résoudre des problèmes

« réservés aux humains » et, point qui nous intéresse particulièrement, d'« utiliser le langage ». La « conférence de Dartmouth » est depuis passée à la postérité et les discussions qui s'y tinrent suscitèrent un optimisme débordant pour la discipline qui y est née : Herbert Simon, dont les travaux sur la modélisation de la prise de décision furent couronnés à la fois par un prix Nobel d'économie en 1978 et un prix Turing¹ en 1975, affirma même que « d'ici 20 ans, des machines seront capables de faire n'importe quel travail actuellement réalisable par un humain »².

Si de nombreuses méthodes ont été envisagées lors de la conférence de Dartmouth, les débuts de l'IA sont marqués par ce qui sera bientôt connu sous le nom d'*approches symboliques*. Ce terme exprime l'idée très simple, clairement énoncée dans le dossier de la fondation Rockefeller, d'obtenir une description suffisamment précise de la manière dont les humains comprennent et emploient le langage pour conférer ensuite à un ordinateur la faculté de traduire des phrases en instructions exécutables. Les participants de la conférence estiment par cette méthode pouvoir même imiter toute autre fonction « intelligente » caractérisant les humains, telle la vision ou le raisonnement. Ainsi, les premiers chercheurs en IA et en TAL ont d'abord cherché à comprendre et à modéliser l'intelligence humaine pour pouvoir la reproduire.

Plusieurs tentatives pour mettre en œuvre cette vision très *mécanique* de la compréhension du langage furent rapidement entreprises. Eliza, un programme développé par Joseph Weizenbaum au MIT entre 1964 et 1966, fut le premier succès remarquable de l'IA dans le domaine de la compréhension et de l'utilisation du langage par un ordinateur. Ce programme jouait le rôle d'un psychothérapeute rogérien (un thérapeute qui retourne souvent les paroles du patient). Voici un exemple de dialogue entre Eliza et une personne :

« Patient : Bonjour

Eliza : Comment allez-vous ? Pourquoi venez-vous me voir ?

Patient : Je me sens triste et fatigué.

Eliza : Pourquoi dites-vous que vous vous sentez triste ?

Patient : Parce que ma compagne ne me comprend pas ?

Eliza : Est-ce que c'est parce que votre compagne ne vous comprend pas que vous êtes venu me voir ?

Patient : Non, c'est ma compagne qui m'a convaincu de vous voir.

Eliza : Plus tôt, vous avez dit que vous vous sentiez fatigué. Pourquoi ? »

Plutôt convaincant si l'on ne connaît pas l'astuce sur laquelle repose le programme !

Pour choisir les questions à poser, Eliza repose sur des règles reproduisant le comportement d'un psychothérapeute humain. Un premier type de règles permet au programme de reformuler sous forme de questions les affirmations du patient. Par exemple, si Eliza détecte « je pense que... » dans une réponse, le programme peut poser la question « pourquoi pensez-vous que... ? » en recopiant la fin de la phrase sans jamais chercher à la comprendre³. Un second type de règles associe certains mots (comme « mère » ou « frère ») à des réponses génériques (« Parlez-moi de votre famille. »). Enfin, un dernier type de règles prend la forme de phrases « type » (« Que voulez-vous dire par là ? », « En êtes-vous sûr ? », etc.) qu'Eliza peut utiliser pour relancer la conversation ou lorsque le programme ne sait pas quoi dire.

Malgré la simplicité de l'approche (il est difficile de voir une quelconque forme d'intelligence ou de compréhension dans un programme qui se fait juste l'écho de ce qu'il vient d'entendre), Eliza est capable de convaincre des humains qu'elle les comprend. Joseph Weizenbaum rapporta de nombreuses anecdotes de personnes pensant que l'ordinateur était devenu leur ami et conclut dans son livre *Computer Power and Human Reason: From Judgment to Calculation*, paru en 1967⁴ : « Je n'avais pas réalisé... que des expositions extrêmement courtes à un programme

informatique relativement simple pouvaient induire de puissantes pensées délirantes chez des personnes tout à fait normales. »

Les réactions des humains interagissant avec Eliza ont permis aux chercheurs de mettre en évidence ce que l'on appelle depuis « l'effet Eliza » : les humains ont naturellement tendance à attribuer une compréhension et une intelligence réelles à un programme informatique, même si celui-ci se contente d'afficher des chaînes de caractères de manière mécanique sans vraiment les comprendre. Ainsi, comme l'explique Douglas Hofstadter⁵, un distributeur de billets concluant une transaction par un « merci » n'est absolument pas capable de ressentir le moindre sentiment de gratitude : c'est le concepteur du distributeur qui aura explicitement programmé la machine pour afficher les caractères correspondant à ce moment précis de la transaction. Pourtant, ce message peut donner l'impression à certaines personnes qu'il y a une personne derrière la machine avec laquelle elles viennent d'échanger. L'effet Eliza, et de manière plus générale l'anthropomorphisme (l'attribution de caractéristiques humaines à des objets ou des animaux), jouent un rôle essentiel dans la manière dont nous interagissons avec ChatGPT. Nous reviendrons plus en détail sur ce point au [chapitre 5](#).

Eliza, première tentative d'interaction en langage naturel avec une machine, a joué un rôle crucial en montrant que ce type d'interaction était possible, à une époque où les ordinateurs étaient souvent perçus comme de simples machines à calculer, tout juste bonnes à faire des opérations arithmétiques. Un bel exploit ! Pour faire fonctionner son programme, Joseph Weizenbaum a introduit des idées essentielles concernant non seulement le développement de ce qui sera bientôt appelé des « systèmes de dialogue » ou « chatbot », mais également la conception de programmes informatiques : pour développer le code d'Eliza, il introduisit les principes d'*abstraction* et de *modularité*, couramment utilisés aujourd'hui encore dans tous les programmes informatiques.

Les préceptes au cœur d'Eliza furent mis en œuvre dans plusieurs systèmes permettant à n'importe quel utilisateur d'interroger des bases de données via le langage naturel, sans avoir à dompter un quelconque langage informatique obscur. Par exemple, Baseball, développé en 1961, donnait le change sur des questions simples sur les matchs de baseball de l'année, telles que « Où les Red Sox ont-ils joué le 7 juillet ? » Un autre programme, Lunar, développé à la fin des années 1960, répondait avec succès à des sollicitations sur l'analyse géologique des roches rapportées de la Lune, par exemple : « Donnez-moi toutes les analyses pour l'hydrogène dans l'échantillon 100 046 ? », ou encore « Quel est le rapport moyen potassium-rubidium dans les basaltes ? »

Le système le plus impressionnant fut sans aucun doute Shrdlu. Ce programme, conçu par Terry Winograd entre 1968 et 1970 au MIT, laissait un utilisateur interagir avec un monde simplifié composé de blocs ayant différentes formes et couleurs, comme dans l'exemple ci-dessous (généré à l'aide de l'outil Povray) :

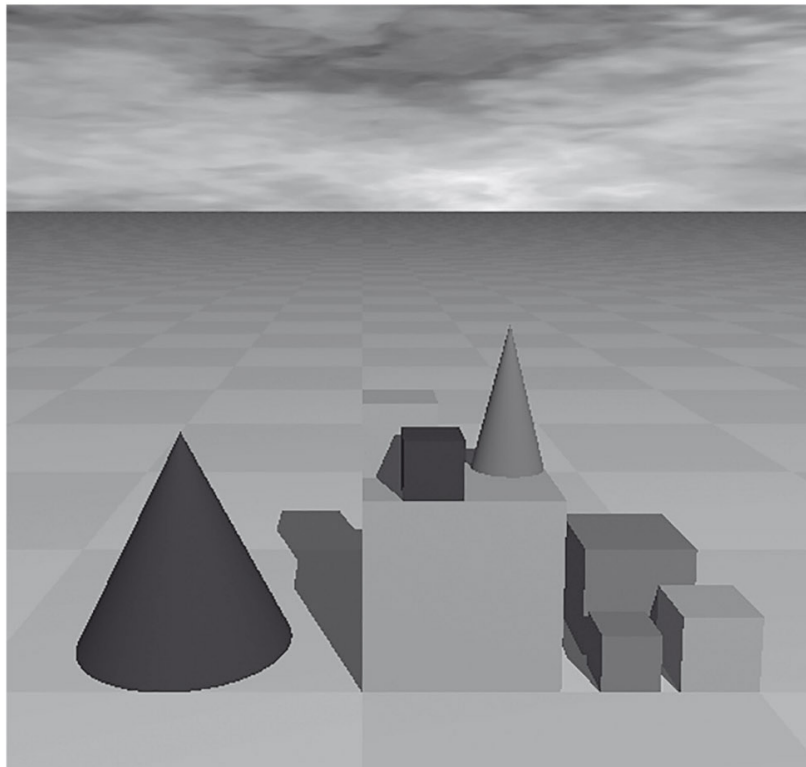


Figure 1. Voici le dialogue entre Shrdlu et l'ordinateur. Shrdlu :
Personne : prend le plus gros bloc gris foncé. Shrdlu : ok. Personne :

pose-le à côté de la pyramide noire. Shrdlu : ok. Personne : Prend un bloc gris clair. Shrdlu : Je ne comprends pas de quel bloc vous parlez. Shrdlu : celui qui est à côté de la pyramide gris foncé. Ordinateur : ok.

Comme l'illustre cet exemple, Shrdlu savait répondre à des questions concernant une scène, exécuter des instructions la modifiant, mais aussi poser les questions nécessaires pour s'assurer de bien comprendre les intentions de son interlocuteur. L'interaction ne se limitait pas à une simple alternance de questions et de réponses : Shrdlu se souvenait de ses actions et des raisons pour lesquelles il les avait accomplies. Il « comprenait » les instructions données et « raisonnait » dessus. Dans son domaine d'utilisation spécifique, Shrdlu n'avait rien à envier à ChatGPT !

Comment fonctionnaient Lunar et consorts ? Un premier module était d'abord chargé d'analyser la phrase pour « comprendre » la question et la convertir en une forme logique facilitant l'interrogation d'une base de connaissances (qui pouvait évoluer, comme dans le cas de Shrdlu). Par exemple, Baseball *traduisait* la question « Où les Red Sox ont joué le 7 juillet ? » en une liste d'informations (toujours les mêmes) permettant d'identifier, pour chaque phrase, les éléments pertinents pour répondre à la question :

Lieu = ?

Équipe = Red Sox

Mois = Juillet

Jour = 7

Il était alors facile, pour un second module, d'interroger la base de données pour retrouver les informations manquantes. Un dernier module utilisait, enfin, des règles similaires à celles d'Eliza pour générer une phrase répondant à la question.

Dans la plupart des cas, il était possible de limiter l'analyse de la phrase à l'identification de mots-clés, comme le faisait Eliza. Par exemple, Baseball comparait le pronom interrogatif en début de

phrase à une liste préétablie pour identifier l'élément sur lequel portait la question : une phrase commençant par *où* ne pouvait porter que sur le lieu d'une rencontre. Le programme disposait également de listes prédéfinies et de règles pour identifier facilement les noms d'équipes, les dates, etc. Il était également possible (bien que plus difficile) d'écrire des règles capables d'effectuer des analyses plus complexes des phrases : Lunar et Shrdlu pouvaient ainsi analyser les subordonnées relatives pour déterminer précisément sur quels éléments portaient les questions et les instructions.

Ces programmes à base de règles, également appelés « systèmes experts », suscitèrent un énorme enthousiasme et furent appliqués à de nombreux problèmes, souvent avec succès, prouvant que les ordinateurs pouvaient comprendre et raisonner à partir d'informations fournies en langage naturel. Pourtant, même dotés de ces qualités, tous les systèmes dont nous avons parlé sont très limités : Lunar était spécialisé dans la géologie (et uniquement celle de la Lune), Baseball dans les parties de baseball (et ignorait les joueurs !) et, de par sa conception, Shrdlu ne fonctionnait que dans un monde artificiel. Aussi, malgré l'engouement des débuts, il apparut assez rapidement que les principes au cœur de ces systèmes se cantonneraient toujours à des domaines très limités, ce que l'on appelle des « mondes clos ». La traduction automatique offre un bel exemple de ces désillusions.

Des règles pour traduire

Dans les années 1960, en pleine guerre froide, la recherche en traduction automatique avait connu un essor sans précédent, stimulée par des besoins militaires et de renseignement. La capacité à traduire et à analyser de vastes quantités de données pour décrypter rapidement des informations potentiellement cruciales pour la sécurité nationale était devenue une priorité scientifique majeure. L'ARPA⁶, l'agence du département de la

Défense des États-Unis chargée des nouvelles technologies militaires, avait décidé de financer plusieurs groupes de recherche pour développer les premiers moteurs de traduction automatique. Ces moteurs utilisaient tous une approche basée sur des règles.

Comme avec Lunar et Baseball, il est relativement simple d'écrire des règles (et donc des programmes) capables de traduire des phrases dans des cas bien définis. Par exemple, pour traduire en français la phrase « *the black cat sleeps* », il suffit d'utiliser un dictionnaire pour lister toutes les traductions possibles des mots de la phrase (« *black* » se traduit en « noir », « noire » ou « noirs », etc.). Ensuite, en utilisant des règles d'accord, il est possible de choisir les formes correctes des articles et des adjectifs. La dernière étape consiste à remettre chaque mot à sa place en se souvenant que, en français, les adjectifs de couleur se placent après le nom (ce qui nécessite de savoir que « noir » est un adjectif de couleur). Toutes ces phases peuvent être décrites assez facilement par des règles génériques exécutables par un ordinateur.

Bien que ce principe soit très simple, sa mise en œuvre pose rapidement des problèmes lorsque les phrases deviennent plus complexes. Cette complexité peut être liée à la richesse du vocabulaire, à la diversité des sujets et des domaines ou aux structures grammaticales utilisées. Non seulement le nombre de règles à prendre en compte augmente rapidement (ce qui soulève la question de savoir dans quel ordre elles doivent être appliquées), mais elles deviennent également de plus en plus difficiles à formaliser.

Par exemple, décrire les règles de placement des adjectifs en français (un problème pourtant relativement simple) de manière suffisamment précise pour les traduire en programme informatique est déjà un défi qui exige des connaissances approfondies de la langue et des ressources considérables (il ne suffit pas de savoir qu'un mot est un adjectif, il faut également

connaître ses *propriétés*) : si aucun locuteur français natif n'hésitera entre dire « un joli petit chat » ou « un petit joli chat », peu de personnes connaissent la règle de placement afférente (en français, les adjectifs décrivant une opinion sont généralement placés avant ceux décrivant la taille).

Malgré ces difficultés, de nombreux efforts ont été déployés pour créer des systèmes de règles, certes de plus en plus sophistiqués, mais réservés à des usages particuliers : chatbot pour la réservation de billets, assistants virtuels pour faciliter l'utilisation de logiciels ou de sites web, etc. Le temps nécessaire pour développer ces applications et leur qualité souvent très décevante contrastent très fortement avec l'optimisme et l'enthousiasme des participants de la conférence de Dartmouth. Portés par les premiers succès de l'informatique, ceux-ci ont très certainement ignoré les spécificités du langage humain, pourtant identifiées depuis longtemps par les linguistes et les philosophes.

Parler, une tâche si simple et pourtant si compliquée

Parler, écouter, lire... nous faisons constamment appel à notre capacité à comprendre et à utiliser le langage avec une aisance déconcertante, comme si c'était aussi simple que de respirer ou de marcher. Et pourtant ! La compréhension des quelques taches d'encre formant les mots que vous venez de lire implique de nombreux processus complexes dont nous n'avons presque jamais conscience. Le langage humain a en effet des caractéristiques uniques qui le distinguent d'autres formes de communication, notamment la communication animale. À cause de ces caractéristiques, le sens d'une phrase ne peut généralement pas être déterminé directement et n'est accessible qu'après une analyse complexe de la phrase.

L'une des premières particularités du langage humain, source de nombreuses difficultés pour la machine, réside dans sa nature discrète et symbolique. Dans un texte écrit², l'association entre un

mot et son sens est arbitraire : une suite de 4 caractères comme le mot « bleu » évoque chez chacun d'entre nous énormément d'informations liées à la fois à la représentation mentale des concepts correspondants, mais aussi à nos souvenirs (la fameuse madeleine de Proust). Cependant, pour une machine (ou une personne ne parlant pas la langue), il ne s'agit que de simples symboles à partir desquels il est quasiment impossible d'inférer la moindre propriété ou information sur le concept associé. Ainsi, dans les mots « coeruleum », « azur » et « majorelle », rien n'indique que ces suites de lettres désignent trois noms de couleurs et sont donc, d'une certaine manière, *reliées*.

La nature discrète des symboles est une spécificité du TAL qui le distingue des données manipulées dans les autres domaines de l'IA. En vision, par exemple, une couleur peut être représentée par l'« intensité » de chacune des trois couleurs primaires sur une échelle de 0 à 255. Il est ainsi possible de définir que, dans le coeruleum, le rouge a une intensité de 0, le vert de 85 et le bleu de 170. Pour l'azur, ces intensités sont de 0, 127 et 255, et pour le majorelle, 96, 80 et 220. Cette représentation fait apparaître directement, aussi bien aux yeux d'un humain que d'une machine, que ces trois couleurs sont des nuances de bleu (la couleur primaire la plus intense), une information qui ne peut être qu'*apprise* pour les mots et qui ne se déduit pas de la lecture des caractères les composant.

Une autre particularité du langage humain est sa nature compositionnelle : les lettres forment des mots qui constituent eux-mêmes des phrases, des règles grammaticales restreignant les combinaisons possibles (on peut dire « le ciel est bleu », mais pas « ciel le bleu est »). Le sens d'une phrase se déduit directement de deux informations : le sens des mots qui la composent et les relations entre ceux-ci qu'il faut inférer à partir des règles syntaxiques. Cette propriété est essentielle. C'est en effet elle qui apporte une réponse à l'une des questions fondamentales de la linguistique, à savoir la possibilité de générer une infinité de

phrases et de sens à partir d'un nombre fini de symboles : s'il est plus que probable que vous ayez déjà vu au moins une fois chacun des mots de la phrase précédente, il est encore plus probable que ce soit la première fois que vous les voyiez ensemble et dans cet ordre. Et pourtant, vous n'avez eu aucune difficulté à en construire le sens !

La compositionnalité soulève plusieurs problèmes. Le premier est lié à ce que l'on appelle le problème des « données creuses » (*sparse data*). Comme les mots peuvent être combinés d'une infinité de manières, il est impossible qu'un ordinateur puisse comprendre le langage uniquement en observant et en mémorisant toutes les utilisations possibles des mots : il faudra que, comme un être humain, il soit capable de construire le sens d'un énoncé à partir de l'observation des mots le composant.

Cette construction du sens se heurte justement à un obstacle : la manière dont les mots et les groupes de mots se combinent n'est jamais explicite et doit donc être déduite. Malheureusement, il existe systématiquement plusieurs façons de combiner les mots, chacune ayant un sens différent. Cette *ambiguïté* du langage se retrouve à tous les niveaux d'analyse d'un énoncé. Par exemple, une phrase aussi simple que « les cris du corps enseignant » a exactement la même prononciation que « les cris du corps en saignant » ou « l'écrit du Coran saignant ». De manière similaire, si je dis « j'ai mangé une pizza avec des anchois », tout le monde (ou presque) m'imaginera devant une magnifique pizza couverte d'anchois, et non pas en train d'essayer de découper ma pizza avec des anchois (ce qui aurait été l'interprétation la plus plausible si j'avais dit « avec des couverts »), ni même attablé avec des anchois (l'interprétation correspondant à « je mange une pizza avec des amis »).

Si aucun être humain n'a de difficulté à identifier parmi ces trois sens possibles le plus plausible, ce choix est en général souvent inconscient et difficile à expliquer. Au contraire, un ordinateur ne

manipulant que des suites de symboles selon des règles précises énumérera très facilement tous les scénarios envisageables, mais aura énormément de difficultés à trancher parmi eux. Cette confusion explique en grande partie les problèmes rencontrés dans la mise au point des systèmes de règles.

Construire automatiquement le sens des phrases

Comment implémente-t-on l'analyse du langage dans un logiciel ? Pour prendre en compte toutes ces spécificités, les spécialistes du TAL ont construit des « chaînes de traitement » (*pipeline*) constituées de modules empilés les uns à la suite des autres qui offrent une représentation de plus en plus abstraite d'une phrase⁸. À la fin, on aboutit à une traduction de la phrase sous une forme logique pouvant être manipulée directement (au moins en théorie) par un ordinateur. La figure ci-dessous donne un exemple d'une telle séquence d'opérations.

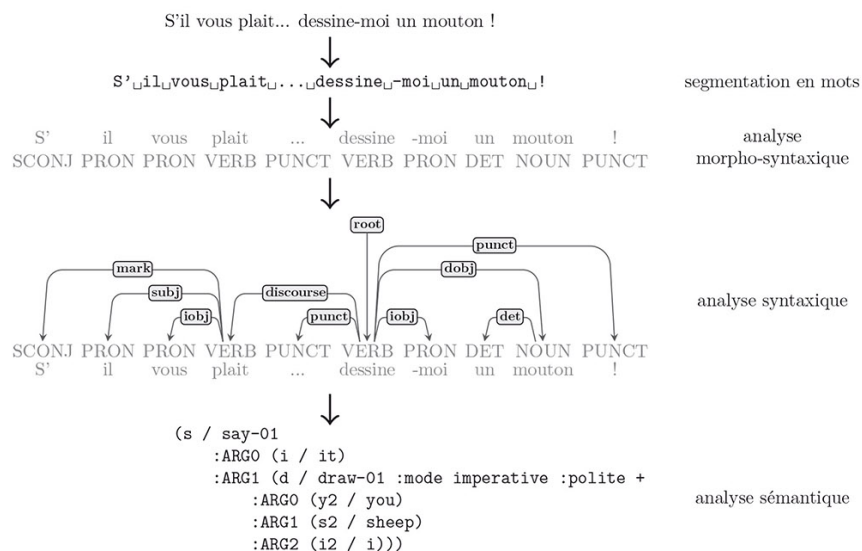


Figure 2. Exemple d'une chaîne de traitement en TAL. La phrase est tout d'abord segmentée en mots. Une première analyse permet alors d'identifier la catégorie grammaticale de chaque mot. Ces informations permettent ensuite de prédire la syntaxe de la phrase puis une représentation de son sens.

Cette chaîne se compose de quatre étapes qui reproduisent (plus ou moins) la manière dont les linguistes modélisent l'analyse

d'une phrase. La première étape – appelée « segmentation en mots » (*tokenisation*) – consiste à découper une phrase en mots. Pour un ordinateur, un texte est simplement une série de caractères, et il est nécessaire de lui expliquer clairement comment identifier les mots qu'il doit manipuler pour comprendre le sens de la phrase. Par exemple, en segmentant la phrase « J'aime le chocolat. », on indique à un ordinateur qu'elle est composée de cinq mots (« j' », « aime », « le », « chocolat », sans oublier le point final, considéré un mot) et non de 19 caractères. Cette étape peut sembler triviale au premier abord et en effet, quelques règles utilisant la ponctuation et les espaces suffisent à traiter la plupart des cas.

Cependant, comme c'est souvent le cas en TAL, de nombreuses exceptions compliquent fortement la tâche : si l'apostrophe indique une frontière de mot dans « j'aime », ce n'est pas le cas dans « rock'n'roll », sans parler de tous les problèmes liés à la segmentation des hashtags (faut-il considérer « #jesuischarlie » comme un mot ou comme plusieurs ?) ou des unités et des nombres.

Chaque mot est ensuite associé à une étiquette morphosyntaxique. Cette étiquette permet d'identifier le rôle que joue le mot dans la phrase (est-ce un nom, un adjectif ou un verbe ?) et parfois également de repérer certaines propriétés morphologiques (est-il est au singulier ? au féminin ? ...). Cette étape a pour objectif essentiel de faciliter l'étape suivante en levant certaines ambiguïtés. Elle permet, par exemple, de distinguer les différentes utilisations possibles de « la », qui peut être un déterminant (« la porte d'entrée »), un pronom (« je la porte ») ou un nom (« je donne le la »). Comme illustré sur la figure [ici](#), après cette étape, chaque mot identifié lors de la première étape est associé avec une étiquette décrivant le rôle grammatical du mot dans la phrase. Ainsi, les mots « il », « vous » et « moi » sont associés à l'étiquette PRON décrivant les pronoms, le mot « si » à SCONJ décrivant les

conjonctions de subordination, et « mouton » à NOUN décrivant un nom⁹.

L'étape suivante consiste à déterminer la structure syntaxique de la phrase afin de visualiser les relations entre les mots la composant. Cette étape soulignera les différents éléments constituant la phrase (par exemple, les frontières d'une subordonnée relative), mais également les relations entre eux (quel nom la subordonnée qualifie-t-elle ?).

S'il existe de nombreux formalismes pour décrire la syntaxe d'une langue, il est courant, en TAL, de représenter la syntaxe d'une phrase par un *arbre de dépendances* comme celui qui est représenté dans l'exemple de la chaîne de traitement (voir [figure 2](#)). Cette structure que les informaticiens appellent *arbre* (même si, de l'avis général, le lien avec les arbres qui poussent dans les forêts n'est pas évident) montre comment chaque mot de la phrase est lié aux autres via des *arcs* (représentés ici par des flèches) symbolisant les relations entre mots. Il permet de décrire la structure de la phrase de manière hiérarchique en identifiant le verbe de la principale (noté « root », sur l'illustration), les compléments de celui-ci en regardant les différents arcs partant du verbe principal puis les compléments de ceux-ci (il suffit à nouveau de suivre les flèches), puis les compléments des compléments, et ainsi de suite. Les étiquettes sur les arcs permettent d'identifier les rôles de chaque complément : l'étiquette *dojb* indique les compléments d'objet direct, *iojb* les compléments d'objet indirect, *subj* le sujet, etc.

Enfin, la dernière étape consiste à réaliser une analyse sémantique de la phrase, en utilisant toutes les informations extraites lors des étapes précédentes, afin de représenter son sens de manière cohérente et non ambiguë. Tout comme dans l'analyse syntaxique, il existe de nombreuses méthodes pour décrire le sens d'une phrase. L'une des plus populaires est le formalisme AMR (pour *Abstract Meaning Representation*). Ce formalisme décompose la phrase en unités sémantiques représentant, par exemple, les

entités, les relations ou les événements, puis décrit les liens entre ces unités. Il permet en particulier d'éliminer les spécificités de la phrase, en normalisant les temps verbaux et en regroupant les synonymes sous un même concept. L'exemple d'analyse considéré identifie ainsi les concepts mis en jeu : l'action de dessiner (le premier sens du verbe « *draw* »), le concept de mouton (également le premier sens de « *sheep* ») et le lien entre ces deux éléments. Le résultat de cette analyse est une structure non ambiguë, facilement utilisable par un ordinateur dans diverses tâches de TAL nécessitant une compréhension du sens de la phrase.

De nombreux programmes ont été développés pour mettre en œuvre des chaînes de traitement (ou des parties de celles-ci) en utilisant différentes approches, qu'il s'agisse de systèmes à base de règles, tels que ceux que nous avons décrits au début de ce chapitre, ou d'approches à base d'apprentissage statistique et, notamment, de réseaux de neurones (que nous introduirons dans les prochains chapitres). Le développement de ces programmes a fait appel à de nombreuses connaissances en linguistique afin de définir les formalismes ciblant les différents niveaux d'analyse de la phrase (par exemple, pour en extraire la syntaxe). Il a également nécessité la collecte de nombreuses ressources, typiquement des dictionnaires identifiant certaines propriétés linguistiques des mots.

Jusqu'au succès des méthodes neuronales à la fin des années 2010, une grande partie de la recherche et des développements en TAL s'est concentrée sur la mise au point de programmes capables de réaliser les différentes étapes de cette analyse et d'explicitier la structure linguistique des phrases. Cette approche, reposant sur la construction explicite du sens d'une phrase (ou du moins d'une représentation de celui-ci), était considérée comme une première étape nécessaire au développement d'applications de TAL plus *avancées* pouvant répondre aux besoins de leurs utilisateurs (traduction automatique, systèmes capables de répondre à des questions, etc.).

L'utilisation d'un processus *transparent* a de nombreux avantages : il est possible d'analyser chaque étape menant à la représentation construite, et celle-ci peut être visualisée pour comprendre les décisions que l'ordinateur prendra par la suite. Les défis soulevés dans la conception de programmes capables de construire explicitement le sens d'une phrase sont toutefois gigantesques, et aujourd'hui encore, s'il existe plusieurs logiciels fournissant des chaînes de traitement pour les langues les plus utilisées, ceux-ci continuent de faire des erreurs. Pour le français, par exemple, les analyseurs syntaxiques automatiques ne sont généralement pas capables de prédire correctement plus de 90 % des arcs d'un arbre de dépendances, et encore, uniquement dans les cas les plus favorables des phrases grammaticalement correctes ne contenant pas de constructions grammaticales trop ambiguës. Finalement, les difficultés rencontrées dans la conception de ces programmes ont rapidement poussé les experts en TAL à explorer d'autres approches ne nécessitant pas toutes ces analyses.

Les vecteurs

Les vecteurs sont des outils mathématiques. Ils jouent aujourd'hui un rôle central en TAL et, plus généralement, dans toute l'IA et l'analyse de données. De manière très intuitive, un vecteur de dimension n est un ensemble de n nombres. Ainsi, 17,03 est un vecteur de dimension 1, un vecteur de dimension 2 sera constitué de 2 nombres, par exemple 12 et 27 et noté soit $(12 \ 27)$, soit $\begin{pmatrix} 12 \\ 27 \end{pmatrix}$, un vecteur de dimension 4 sera par exemple $(1 \ 2 \ 3 \ 4)$ et ainsi de suite. La i -ème composante d'un vecteur $\mathbf{x}$, notée x_i , correspond au i -ème nombre le composant. Dans la suite, les vecteurs seront représentés soit par des lignes, soit par des colonnes comme :

vecteur colonne : $\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$ $\longrightarrow$ 

vecteur ligne : $(1 \ 2 \ 3 \ 4)$ $\longrightarrow$ 

Comme nous le verrons, les vecteurs manipulés en TAL sont très souvent creux : la plupart de leurs composantes sont égales à 0. Dans ce cas, nous les représenterons par :

vecteur colonne $\longrightarrow$ 

vecteur ligne $\longrightarrow$ 

Dans cette représentation, les parties foncées correspondent aux composantes non nulles. Il est possible de donner une interprétation géométrique aux vecteurs. Pour les vecteurs de dimension 2, chaque vecteur peut être associé à un point dans un espace à deux dimensions : la première composante correspondra à la position du point sur l'axe des abscisses, la seconde à sa position sur l'axe des ordonnées (voir [figure 3](#), ci-dessous). Si elle ne permet des représentations que pour les vecteurs de dimensions 2 et 3 (notre capacité à percevoir visuellement l'espace étant limitée à trois dimensions), cette interprétation géométrique se généralise aux vecteurs de plus grande dimension et permet d'interpréter intuitivement les opérations sur les vecteurs.

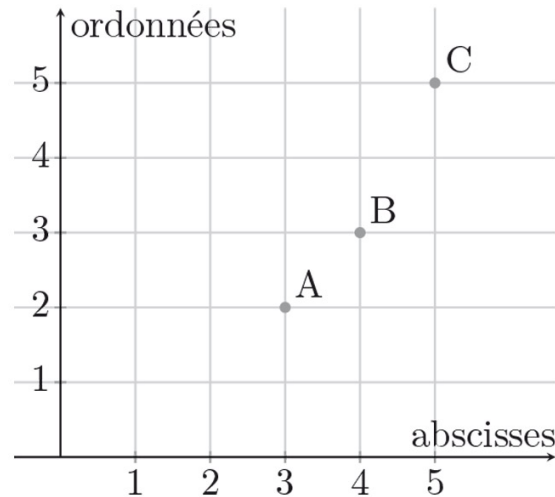


Figure 3. Représentation graphique des vecteurs $(3,2)$, $(4,3)$ et $(5,5)$ par, respectivement, les points A , B et C .

De nombreuses opérations mathématiques sont, en effet, possibles sur les vecteurs (un sous-domaine entier des mathématiques, l'algèbre linéaire, est consacré à leur étude). Il est, par exemple, possible d'additionner deux vecteurs en faisant la somme composante par composante comme dans :

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} = \begin{pmatrix} 1 + 4 \\ 2 + 5 \\ 3 + 6 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 9 \end{pmatrix}$$

Il est également possible de définir des distances ou des similarités entre deux vecteurs pour mesurer à quel point ceux-ci sont différents ou, au contraire, proches.

La notion de distance est une généralisation directe de celle que nous utilisons lorsque nous regardons une carte : sur l'exemple de la figure ci-dessus, la distance entre A et B est plus petite que celle entre A et C , ce qui indique que les composantes des vecteurs représentés par A et B sont plus *proches* que les composantes de A et C . Cette intuition sert de base à la définition formelle de plusieurs notions de distance (par exemple, la distance euclidienne) ou de similarité (par exemple, la similarité cosinus) entre deux vecteurs, qui reposent toutes sur la comparaison des valeurs des composantes des deux vecteurs. Par exemple, la distance euclidienne entre deux vecteurs A et B (utilisée pour mesurer la distance entre deux points sur une carte) se calcule simplement en soustrayant la i^{e} composante de A à la i^{e} composante de B , en additionnant les carrés des différences ainsi obtenues, puis en prenant la racine carrée du résultat.

De manière intuitive, plus les valeurs de deux composantes diffèrent, plus la distance entre les deux vecteurs sera grande.

Compter pour comprendre

Une approche alternative, ne reposant pas sur la construction explicite du sens, le modèle vectoriel, a joué un rôle crucial dans le développement de la capacité des ordinateurs à comprendre le langage naturel. Dans ce modèle, les mots et les documents sont représentés par des nombres, plus précisément par des ensembles de nombres, que l'on peut représenter par des vecteurs (voir [encadré](#)).

Cette approche a principalement émergé au début des années 1960, au sein d'un domaine étroitement lié au TAL, la recherche d'information (RI), tout en ayant ses racines dans des travaux antérieurs en psychologie et en linguistique. La RI est un domaine scientifique dont l'objectif central est le développement de moteurs de recherche tels que Google, Bing, ou DuckDuckGo.

La RI et le TAL sont deux domaines historiquement distincts, mais partagent de nombreux objectifs similaires. Ils visent tous deux à permettre aux ordinateurs de *comprendre* des textes afin de répondre aux questions des utilisateurs. Cependant, malgré ces objectifs communs, ces deux domaines ont traditionnellement adopté des approches différentes. Le TAL se concentrait sur la construction d'une représentation explicite du sens des phrases, tandis que la RI utilisait souvent les méthodes basées sur le comptage des mots que nous allons présenter dans cette section. Avec l'avènement des réseaux de neurones, sur lesquels nous reviendrons au [chapitre 3](#), une convergence significative s'est opérée entre ces deux domaines.

La tâche principale de la RI correspond exactement à ce que nous faisons tous les jours lorsque nous cherchons des informations en ligne : à partir d'une requête, le moteur de recherche tente

d'identifier parmi un ensemble de documents (un corpus) ceux qui sont pertinents vis-à-vis de la demande. Dans la plupart des cas, le corpus est constitué de l'ensemble des pages web accessibles sur l'Internet. Pour accomplir cette tâche, le programme doit déterminer si un document (qui peut contenir seulement quelques mots ou, au contraire, de nombreuses pages) et une requête (généralement une phrase) traitent du même sujet, ce qui revient, d'un point de vue technique, à mesurer la similarité sémantique entre ces deux éléments.

Le modèle vectoriel suppose que le vocabulaire, c'est-à-dire l'ensemble des mots pouvant apparaître dans un texte, soit connu et qu'il ne change pas (un mot qui n'est pas connu à l'instant précis de la conception du programme sera, la plupart du temps, simplement ignoré). Cette hypothèse est, bien sûr, irréaliste pour de nombreuses applications et nous verrons au [chapitre 3](#) comment les systèmes actuels parviennent à gérer les nouveaux mots.

Comme nous l'avons vu, les mots sont des entités discrètes. Chaque mot peut donc être associé de manière arbitraire à un nombre unique, compris entre 1 et N , où N est la taille du vocabulaire pouvant lui servir d'*identifiant*. Les différentes occurrences d'un même mot dans un corpus auront toutes le même identifiant, comme dans l'exemple de la figure ci-dessous.

Phrase 1	Rodrigue	,	qui	l'	eût	cru	?
	↓	↓	↓	↓	↓	↓	↓
	6	1	5	4	9	8	2
Phrase 2	Chimène	,	qui	l'	eût	dit	?
	↓	↓	↓	↓	↓	↓	↓
	3	1	5	4	9	7	2

Vocabulaire			
,	1	Rodrigue	6
?	2	dit	7
Chimène	3	cru	8
l'	4	eût	9
qui	5		

Figure 4. Exemple d'encodage d'un corpus composé de deux phrases.

Cet exemple montre que les associations sont complètement arbitraires : les identifiants de deux noms, même si ceux-ci jouent des rôles similaires (il est possible de remplacer l'un par l'autre en gardant une phrase grammaticalement correcte et compréhensible), *Rodrigue* et *Chimène*, n'ont aucun lien.

Dans cet exemple, très simplifié, il n'y a que deux textes, constitués chacun d'une unique réplique. La taille du vocabulaire est donc très réduite. Dans des applications réelles, il n'est pas rare de considérer des vocabulaires comportant plusieurs centaines de milliers de mots différents. Un corpus aussi modeste que celui constitué des 2,4 millions de pages de la Wikipédia francophone contient déjà près de 6,5 millions de mots différents après une segmentation en mots naïve (il ne faut pas oublier que la notion de mot inclut toutes les dates, les nombres, les noms propres, les sigles, etc.). Ce nombre est beaucoup plus important pour les langues ayant une morphologie complexe (typiquement celle avec des déclinaisons) ou les langues agglutinantes¹⁰.

Ces identifiants permettent également de représenter un mot par un vecteur ayant une dimension égale à la taille du vocabulaire, avec toutes les composantes nulles à l'exception de celle

correspondant à l'identifiant du mot. Ainsi, dans l'exemple précédent, le mot *Chimène* sera représenté par un vecteur de dimension 9 (le nombre de mots différents dans le corpus) :

$$\text{Chimène} = \begin{pmatrix} 0 & 0 & \text{1} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

?
l'
Rodrigue
cru

↓
↓
↓
↓

↑
↑
↑
↑

,
Chimène
qui
dit
eût

La i -ème composante de ce vecteur est associée au mot (indiqué en gris dans l'illustration ci-dessus) dont l'identifiant dans le vocabulaire est i . La valeur de cette composante est soit 1 (si la position correspond à l'identifiant du mot dans le vocabulaire), soit 0 (dans le cas contraire).

Cette représentation sous forme de vecteurs, appelée *one-hot*, permet de construire facilement des représentations de texte, que ce soit une phrase, un paragraphe, un document ou toute autre unité de texte, en additionnant les vecteurs représentant les mots qui le composent. Ainsi, en poursuivant l'exemple précédent, la réplique « qui l'eût cru » sera représentée par le vecteur suivant :

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

qui
l'
eût
cru

Comme déjà souligné, il ne faut pas perdre de vue que les exemples de représentations donnés ici sont simplifiés à l'extrême : dans des applications réelles, les textes sont représentés par plusieurs centaines de milliers de nombres, la plupart (tous

ceux correspondant aux mots n'apparaissant pas dans le texte) étant égaux à 0.

Intuitivement, cette représentation d'un texte correspond au nombre d'occurrences, appelé « tf » (pour *term frequency*), de chaque mot du vocabulaire dans celui-ci : si un mot apparaît 3 fois dans un texte, la composante correspondant à son identifiant sera égale à 3 dans le vecteur représentant le texte. Cette représentation est souvent qualifiée de « sac de mots » (*bag of words*) pour souligner le fait qu'elle ne repose que sur l'identification des mots présents sans tenir compte de la structure (typiquement syntaxique) des phrases. Ainsi, les deux phrases « Nour aime Camille » et « Camille aime Nour » auront exactement la même représentation, même si la personne dont les sentiments sont mentionnés n'est pas la même. Bien qu'elle permette d'identifier que la phrase parle de Nour et d'amour (ce qui est suffisant pour répondre à la plupart des requêtes), elle ne permet pas de répondre à des questions plus fines sur le sens de la phrase, par exemple pour savoir qui est amoureux.

Représenter les textes par des vecteurs permet d'utiliser différents outils mathématiques pour estimer la similarité ou la distance entre deux textes. C'est une tâche essentielle pour la RI. Mais mesurer automatiquement la similarité entre deux textes fournit également un outil très puissant pour les chercheurs en sciences humaines (voir [encadré ci-après](#), qui montre, par exemple, comment ces méthodes peuvent être utilisées pour comparer les discours de deux hommes politiques).

Analyser les discours politiques

Les représentations tf.idf ont longtemps été la méthode reine pour déterminer automatiquement le sens d'un texte. Mais elles ont également été employées par des chercheurs dans de nombreux domaines tels que la littérature, la sociologie, ou encore les sciences politiques, pour

analyser des documents et extraire des informations utiles à leurs recherches.

Par exemple, il est possible d'utiliser ces représentations pour comparer les vœux des Présidents de la République. Ces discours sont généralement l'occasion pour le Président de dresser un bilan de la situation du pays et, surtout, d'établir les perspectives pour l'année à venir, reflétant ainsi ses idées politiques et ses objectifs. Ils contiennent donc, intuitivement, de nombreuses informations pertinentes pour caractériser la politique de ceux qui les prononcent.

Les discours les plus récents (depuis 1974) sont accessibles en ligne¹¹. Il est relativement facile, avec des connaissances de base en informatique, de télécharger l'ensemble de ces documents et de calculer les représentations tf.idf du corpus ainsi constitué.

Grâce à ces représentations, on peut ensuite identifier les mots les plus représentatifs de chaque discours (ceux avec le score tf.idf le plus élevé). Par exemple, le tableau suivant reprend les 5 mots les plus significatifs des vœux de Nicolas Sarkozy en 2009, 2010 et 2011. Ceux-ci montrent clairement le rôle important que tenait la crise financière de 2008 dans ces discours et l'évolution de celle-ci :

2009	2010	2011
financière	devoir	défis
idée	fin	question
supérieur	respect	agence
fiscalité	fut	résisté
promesse	défendra	sujet

Ces représentations peuvent également servir à comparer automatiquement deux discours et à mesurer leur similarité. À l'aide de méthodes de fouille de données (*data mining*), on peut ainsi regrouper les Présidents par proximité des thèmes abordés.

La figure ci-dessous montre les regroupements automatiquement déterminés à partir des représentations tf.idf de notre corpus de vœux présidentiels. Dans ce graphe, chaque cellule traduit la distance entre les discours dont les noms sont indiqués en ligne et en colonne : plus la

cellule est foncée, plus les deux discours sont similaires. Les lignes et les colonnes sont organisées de manière à regrouper les discours les plus proches côte à côte. Ainsi, les vœux de Jacques Chirac en 1997 et en 1998 (représentés dans le coin inférieur droit) sont très proches. Ce regroupement est hiérarchique : une fois les deux textes les plus similaires identifiés, ceux-ci sont « fusionnés » et la mesure entre le nouveau texte obtenu et tous les autres est à nouveau mesurée afin de pouvoir répéter la procédure. La structure présente en haut du graphe décrit les fusions effectuées. Le graphe révèle deux zones de fortes similarités : l'une regroupant les discours de Valéry Giscard d'Estaing, une autre principalement ceux des quatre derniers Présidents (Jacques Chirac, Nicolas Sarkozy, François Hollande et Emmanuel Macron). À l'intérieur de cette deuxième zone, les discours d'Emmanuel Macron sont particulièrement similaires.

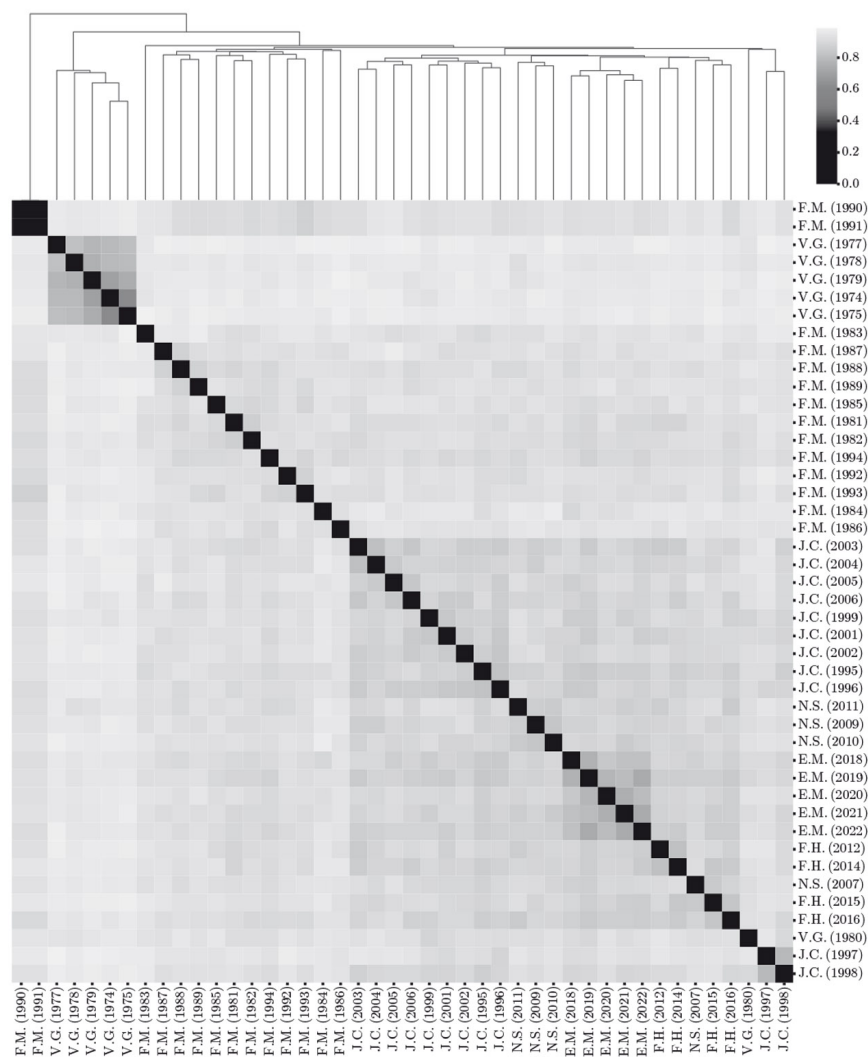


Figure 5. Utilisation des tf.idf pour mesurer la similarité entre les vœux des Présidents de la République.

De plus en plus répandue dans les sciences humaines, l'utilisation de représentations tf.idf (et de nombreux autres outils de TAL) fournit aux chercheurs du domaine des informations pertinentes pour alimenter leurs réflexions. De nouvelles disciplines, telles que les humanités numériques (*digital humanities*) ou le journalisme de données (*data journalism*), se sont développées à l'intersection des domaines traditionnels et de l'informatique. Il n'est donc pas surprenant que de plus en plus de cursus en sciences humaines intègrent des cours d'informatique, d'analyse de données et de TAL !

Il existe plusieurs manières de mesurer la similarité ou la distance entre deux vecteurs. Toutes comparent les valeurs des composantes de même rang (voir [encadré](#)). Ainsi, pour l'exemple de la RI, un texte sera jugé d'autant plus pertinent vis-à-vis de la requête formulée qu'il partagera de mots avec la demande et que ces mots seront fréquents dans le document. Cela revient à considérer que le sens d'un texte est défini directement par les mots qui y sont employés, et que plus un mot y apparaît, plus il contribue au sens du texte. Il est important de bien comprendre que dans cette comparaison, l'ordinateur ne se base que sur la position des mots dans le vecteur (leur identifiant), et non sur le sens de ceux-ci : si une requête contient le mot « chat » et un document le mot « félin » (ou même « chats »), ces mots ayant des identifiants différents ne seront pas détectés comme identiques et ne renforceront pas la similarité entre les deux textes.

Si l'idée d'utiliser la fréquence des mots pour capturer le sens d'un texte est très naturelle, sa mise en œuvre directe ne fonctionne hélas pas. En effet, en étudiant la fréquence à laquelle les mots sont utilisés dans les phrases (ce que l'on appelle la « distribution » des mots), on s'aperçoit qu'un petit nombre de mots, les « mots-outils » (*stop words*) sont très fréquemment utilisés dans des textes, mais ne portent généralement pas de sens spécifique et ne contribuent pas à la compréhension du contenu. Ainsi, si l'on regarde les 20 mots les plus fréquents de la Wikipédia en français (voir [tableau](#) ci-dessous), on voit qu'ils

apparaissent dans quasiment chacune des 2 364 419 de pages de l'encyclopédie¹². La présence de ces mots outils va bruite le calcul d'une similarité basée uniquement sur les comptes : quelle que soit la paire de textes considérée, il est probable que plusieurs mots outils apparaissent dans chacun d'entre eux, augmentant artificiellement leur similarité.

Tableau 1. Les 20 mots les plus fréquents dans la version française de la Wikipédia. Pour chaque mot, sont précisés le nombre d'occurrence du mot, le nombre d'articles dans lesquels le mot apparaît au moins une fois et le pourcentage d'articles que cela représente.

Mot	Nombre occurrences	Nombre documents	Mot	Nombre occurrences	Nombre documents
de	65 351 222	2 355 031 (98,04 %)	est	13 243 838	2 214 424 (92,19 %)
la	30 236 952	2 098 858 (87,38 %)	les	12 190 577	1 397 418 (58,17 %)
et	25 418 838	2 149 734 (89,49 %)	un	10 492 879	1 815 645 (75,59 %)
le	20 749 269	1 982 436 (82,53 %)	une	9 097 061	1 714 326 (71,37 %)
à	20 598 165	1 902 615 (79,21 %)	dans	8 593 690	1 651 224 (68,74 %)
en	19 249 324	2 110 043 (87,84 %)	par	7 729 905	1 426 424 (59,38 %)
l'	16 516 419	1 760 038 (73,27 %)	au	7 130 702	1 511 763 (62,94 %)
du	15 885 832	1 908 936 (79,47 %)	pour	6 239 670	1 165 952 (48,54 %)
des	15 580 490	1 780 587 (74,13 %)	Le	5 346 347	1 223 340 (50,93 %)
d'	13 432 588	1 765 666 (73,51 %)	qui	5 126 682	983 484 (46,31 %)

Une idée très simple permet d'éviter ce problème en incluant, dans la représentation, une seconde information : le nombre de textes du corpus dans lequel un mot est employé. Intuitivement, un mot apparaissant dans tous les textes d'un corpus n'apporte que très peu d'information sur le sens de celui-ci, et surtout, ne permettra pas de le distinguer des autres documents : les termes rares sont plus discriminants et sont plus à même de capturer le sens d'un texte. Il faut donc, pour déterminer l'importance d'un mot, également considérer son idf (*inverse document frequency*), défini comme le rapport entre le nombre de textes du corpus et le nombre de textes dans lesquels le mot apparaît.

Par exemple, en reprenant les informations du [tableau 1](#), l'idf de « la », qui apparaît dans plus de 80 % des articles, est égal à $2\,364\,149$ divisé par $2\,098\,858$, soit 1,12. À l'inverse, le mot « chat », qui n'apparaît que 16 753 fois dans tout le corpus, a un idf de 125,2. Le tf et l'idf sont ensuite multipliés¹³ entre eux pour déterminer le score tf.idf, qui permet de caractériser l'importance relative d'un mot pour déterminer le sens d'un texte : plus ce score est élevé, plus le mot est *représentatif* du sens du document.

La représentation tf.idf permet de résoudre de nombreux problèmes reposant sur l'analyse de textes. Sa principale force réside dans sa simplicité : elle ne nécessite *que* de pouvoir compter le nombre de mots apparaissant dans un corpus, et aucune des analyses plus *profondes* des documents, beaucoup plus coûteuses en temps de calcul, évoquées précédemment. Le choix de cette représentation illustre une des règles d'or du TAL à l'origine de nombreuses décisions dans la conception des modèles : il est souvent préférable de privilégier une solution simple capable de traiter un très grand nombre de textes plutôt qu'une solution plus complexe qui ne peut être appliquée qu'à un corpus de taille limitée.

Cette représentation a été utilisée, comme nous le verrons au chapitre suivant, pour représenter les textes dans de nombreux

systèmes de TAL. Mais c'est dans le cadre de la RI, un domaine où les méthodes économes en temps de calcul sont nécessaires pour pouvoir traiter la masse des documents accessible sur l'Internet, qu'elle a le mieux montré sa puissance : les scores tf.idf ont été, jusqu'à la fin des années 2020, l'une des principales informations¹⁴ utilisées par tous les moteurs de recherche pour déterminer la pertinence d'une page.

La qualité des résultats retournés par les moteurs de recherche montre que, grâce à une utilisation – que l'on pourrait qualifier d'« astucieuse » – d'informations aussi simples que les fréquences d'utilisation des mots, il est possible de *comprendre* si un document répond à une requête. Finalement, comprendre le langage n'est pas si compliqué !

2

APPRENDRE EN FAISANT : LA RÉVOLUTION DE L'APPRENTISSAGE STATISTIQUE

Dans son sens global, l'apprentissage statistique consiste à faire émerger des règles, de façon automatique, seulement à partir d'exemples. Cette méthode a été à l'origine de progrès significatifs en TAL, même si ses limites sont rapidement apparues : elle nécessite de vastes corpus d'exemples étiquetés et implique un travail considérable de préparation. Néanmoins, les premières méthodes d'apprentissage ont jeté les bases théoriques et les éléments fondamentaux qui ont finalement conduit à l'émergence des giga-modèles.

Grâce aux représentations par vecteur d'un texte (qu'il s'agisse d'une phrase, d'un paragraphe ou d'un document entier), un ordinateur sait en faire émerger les thèmes et, dans une certaine mesure, le sens global. Toutefois, il faut encore une intervention humaine pour que ces représentations soient ensuite transformées en *décisions* (ce texte est-il une question ? un spam ?) ou en *actions* (pour traduire un texte, le résumer ou répondre à une question posée). Après l'idée, que l'on peut qualifier sans trop d'hésitation de révolutionnaire, qui a remplacé les mots (des symboles) par des vecteurs, une deuxième révolution du TAL, l'apprentissage automatique¹, a été nécessaire pour conférer aux ordinateurs les mêmes facultés d'initiative, et ainsi franchir un pas vers des systèmes *intelligents* comme ChatGPT.

L'apprentissage peut être vu, en première approximation, comme un moyen de construire automatiquement une fonction, c'est-à-dire une partie d'un programme informatique qui accomplit une tâche spécifique : traduire une phrase, trouver le moyen le plus rapide d'atteindre une destination, trier ou inverser une liste de nombres, etc. Comme une fonction mathématique, son homologue informatique reçoit en entrée des données et renvoie une sortie correspondant au traitement effectué.

L'apprentissage est une réponse directe aux limites des systèmes de règles et, plus généralement, de l'IA symbolique que nous avons décrits dans le chapitre précédent : plutôt que de laisser le programmeur décrire les étapes nécessaires pour transformer ou analyser un texte, l'idée est de fournir à un algorithme d'apprentissage un ensemble d'exemples de ce traitement. Ce *corpus d'apprentissage* est constitué de deux types d'éléments : des *observations* correspondant à des textes représentatifs des entrées que devra être capable de traiter la fonction et, pour chacune de ces observations, la réponse qui devra être *prédite* par la fonction. Ce corpus d'apprentissage est généralement constitué de nombreux exemples choisis au hasard, sans chercher à assurer la moindre représentativité.

L'objectif d'un algorithme d'apprentissage est alors d'*inférer*, à partir de cet ensemble d'apprentissage, une fonction capable de réaliser ce traitement sur n'importe quel texte. Par exemple, un algorithme d'apprentissage pourra être utilisé pour construire automatiquement une fonction traduisant en anglais (la sortie de la fonction) n'importe quelle phrase en français (son entrée) à partir d'un vaste corpus de phrases en français associées à leur traduction en anglais. Il est important de noter, dès à présent, le rôle central que jouent les données et, plus précisément, les données annotées, c'est-à-dire associant des observations aux réponses attendues : c'est de ces associations que l'algorithme d'apprentissage va extraire les informations dont il a besoin pour

construire la fonction. Pour bien mettre en évidence le rôle des annotations, l'apprentissage est qualifié de *supervisé*.

Une des motivations de l'apprentissage est qu'il est souvent beaucoup plus simple et rapide de donner la sortie correspondant à une entrée que d'expliquer et formaliser en instructions les différentes étapes permettant de l'obtenir. Ainsi, en reprenant l'exemple de la traduction, il est extrêmement facile (avec un peu d'entraînement) de traduire une phrase en anglais alors qu'expliquer, de manière suffisamment générale et formelle pour être *traduite* en règles, les raisons derrière le choix de chaque mot ou groupe de mots est une tâche particulièrement chronophage et complexe. L'apprentissage s'apparente donc à une approche de type *montrer et ne pas expliquer*².

L'apprentissage est construit autour de la notion de *généralisation* : son objectif fondamental est de trouver, à partir de l'analyse d'un ensemble *fini* d'exemples, une fonction capable de traiter un ensemble *infini* d'observations. Pour cela, il est nécessaire que l'algorithme extraie les *connaissances* du corpus d'apprentissage qui lui permettront de traiter de nouvelles entrées. Cette contrainte est essentielle : l'algorithme d'apprentissage ne doit pas copier les décisions observées sur quelques textes spécifiques (ceux de l'ensemble d'apprentissage), il faut qu'il extraie des informations *abstraites* qui serviront ensuite sur tous les textes imaginables. Il doit donc pouvoir inférer une règle générale (par exemple, que le verbe s'accorde en nombre avec son sujet) plutôt que de mémoriser un cas particulier (par exemple, que le groupe nominal « les chats » est suivi par « miaulent » et non « miaule »).

L'apprentissage est une approche particulièrement adaptée pour traiter les problèmes dans lesquels les décisions reposent sur des critères multiples, flous et difficiles à formaliser, comme la plupart des problèmes faisant intervenir du texte ou tout autre type de données *brutes* (audio, image, etc.). Ce n'est toutefois pas une méthode générique, et pour de très nombreux problèmes on

préfère programmer explicitement les différentes étapes nécessaires à un traitement. Il n’y a, par exemple, aucun intérêt à appliquer ce principe à une fonction de calcul d’impôts à payer ou de salaires à percevoir. Dans ces deux cas, le chiffrage repose sur des règles très bien définies, par opposition à celles qui régissent le choix de la traduction d’un mot, si difficiles à formaliser que le choix change souvent d’un traducteur à un autre.

« À voir » ou « à jeter » : déterminer la polarité d’un texte

Pour illustrer les différents aspects de l’apprentissage, nous allons nous intéresser à une nouvelle tâche de TAL : l’analyse de sentiments, également appelée analyse d’opinion ou analyse de polarité. L’objectif de cette tâche, dans sa formulation la plus simple, est de construire une fonction capable de déterminer si un texte (par exemple un commentaire sur un film ou sur un produit) exprime un avis positif, neutre ou négatif. Cette fonction doit, par exemple, être capable d’identifier que contrairement au premier commentaire du [tableau 2](#), le second porte un regard très critique sur le film qu’il analyse³. Il s’agit d’une tâche de classification, l’un des problèmes les plus fondamentaux en TAL dont l’objectif est de choisir, parmi un ensemble prédéfini d’étiquettes possibles, celle qui correspond le mieux à un texte donné. Dans le cas de l’analyse de sentiment, nous pouvons ainsi définir trois étiquettes : *positif*, *négatif* et *neutre*. Les fonctions apprises sont alors appelées des *classifieurs*, et l’ensemble des exemples ayant une même étiquette une *classe*.

L’analyse de sentiments a de très nombreuses applications. Elle a, par exemple, été utilisée pour déterminer l’impact d’une campagne de publicité (en cherchant à quantifier si le nombre d’avis positifs a augmenté après la diffusion de la campagne)⁴, mais aussi pour observer la diffusion des opinions dans les

réseaux sociaux⁵ ou pour mettre à nu la construction des intrigues dans les romans⁶.

Comme le montrent les exemples du tableau 2 (voir [ici](#)), il est assez facile de déterminer la polarité d'un texte en identifiant quelques mots saillants. Ainsi, dans le premier exemple, les mots « plaisir », « super » ou « réussie » sont des indications claires que le commentaire est positif. La détection de ces mots n'est toutefois pas un critère suffisant pour jauger de quel côté de la balance penche le commentaire : il est probable qu'une même critique (comme le troisième exemple du [tableau 2](#)) contienne à la fois des mots à connotation positive (dans ce cas « prenant » et « réussite ») et négative (« ennuyer » et « endormir »). Aussi le classifieur devra-t-il combiner les différentes informations présentes avec un certain *doigté*.

L'analyse de sentiments soulève également certaines difficultés spécifiques à cette tâche. La première est la gestion des négations : ainsi, dans le troisième commentaire, l'expression « grande réussite » est niée et n'indique donc pas une opinion positive. Autre point délicat, l'utilisation fréquente de l'ironie : par exemple, dans le dernier commentaire, le défaut mentionné (la durée trop courte du film) n'en est pas vraiment un. La présence de certains mots n'est donc pas un critère suffisant pour conclure sur l'opinion exprimée dans un texte. Au final, l'analyse de sentiments est symptomatique d'un problème à décisions floues et multicritères qui appelle une solution reposant sur l'apprentissage.

Tableau 2. Extrait d'un corpus d'analyse de sentiments associant des commentaires produits par les utilisateurs du site AlloCiné à une étiquette indiquant si le commentaire émet un avis positif (😊) ou négatif (😞).

Étiquette	Commentaire
	On retrouve avec plaisir nos deux ripoux préférés pour une

😊	super histoire encore. Cette suite est très bien réussie !
😐	Un film assez lent et long, et l'on s'ennuie devant certains plans que l'on juge facilement inutiles. L'histoire, également, n'est pas forcément très captivante. Malgré tout, il y a des instants de poésie, la photographie des grands espaces américains est belle, et les acteurs sont convaincants.
😞	Tout est fait dans ce film pour ennuyer et endormir le spectateur. Si son remake <i>Dragon rouge</i> n'est pas une grande réussite, il est toutefois plus prenant que ce <i>Sixième Sens</i> .
😊	Tout commence comme une comédie légère avant un drame soudain qui fait basculer le film dans un... suspense incroyable mêlé d'une étude psychologique très fine des mécanismes qui régissent un couple, en particulier, et un groupe social, en général. [...] Interprété de façon magistrale (mention spéciale aux actrices, sublimes), <i>À propos d'Elly</i> est un conte persan moderne auquel on ne voit qu'un seul défaut : ne durer que 116 minutes.

La conception d'un système capable de prédire la polarité d'un texte via l'apprentissage statistique repose sur quatre étapes que l'on retrouve dans toutes les approches à base d'apprentissage statistique : la collecte d'un corpus de données étiquetées, le choix d'une représentation des informations, la sélection d'une classe de fonction et, finalement, l'apprentissage à proprement parler. Nous allons maintenant détailler chacune de ces étapes.

Les données : l'« électricité de demain »^Z

L'apprentissage supervisé, rappelons-le, repose sur l'analyse (automatique) d'un corpus annoté comme celui du [tableau 2](#), qui associe des exemples d'entrées à une étiquette. Il existe, aujourd'hui, deux manières de créer un tel corpus : recruter des personnes dont le seul travail sera de renseigner à la main les

informations nécessaires, ou se tourner vers les données d'anonymes publiées sur des différentes plateformes du web (TikTok, Youtube, Facebook, etc.).

Cette étape est, en général, la plus longue, la plus difficile et la plus coûteuse de l'apprentissage, même si son importance tend à être négligée. L'[encadré](#) décrit de manière plus détaillée ces deux méthodes ainsi que les coûts et les enjeux, notamment sociaux, liés à la collecte des données.

Construire des représentations

Une fois un corpus annoté collecté, une première étape consiste à transformer chaque exemple en un *vecteur de caractéristiques*. Les classifieurs ne manipulent en effet jamais directement les textes, images ou autres *objets* sur lequel ils sont appliqués, mais des *représentations* qui prennent la forme de vecteurs. Dans le cas des problèmes de classification, les représentations des textes décrites au chapitre précédent (par des sacs de mots ou des vecteurs tf.idf, voir p. 41) sont souvent adoptées. C'est pourquoi, pour la suite, j'ai conservé cette approche ; chaque commentaire sera donc décrit par un vecteur x de n composantes, chaque composante indiquant l'absence ou la présence d'un mot.

Si ce *modus operandi* est adapté à l'analyse des sentiments, une tâche dans laquelle les mots utilisés jouent un rôle important, rien n'empêche des représentations plus riches. Il est tout à fait possible d'ajouter une composante indiquant si le commentaire a été publié pendant la nuit ou non, quel est le temps majoritairement employé dans le texte, si celui-ci contient des fautes d'orthographe et quelle est la longueur moyenne des phrases : il n'y a aucune contrainte, mis à part l'imagination du programmeur, sur le type d'information pouvant être encodé. La seule limite est la taille du corpus d'apprentissage : plus le nombre de caractéristiques est élevé, plus il faudra fournir des observations et leurs étiquettes à l'algorithme d'apprentissage.

Cette transformation des textes en vecteurs, fixée dès la conception du programme, joue un rôle essentiel dans l'apprentissage : elle permet de définir, avant même le choix et la mise en œuvre d'un algorithme d'apprentissage, une représentation adaptée à la tâche. Le classifieur n'a alors plus qu'à se concentrer sur la prise de décision plutôt que la recherche de la représentation.

Définir une classe de fonctions

Une fois la représentation des exemples déterminée, la dernière étape consiste à choisir à la fois une famille de classifieurs et une méthode d'apprentissage offrant le meilleur résultat espéré. En effet, l'apprentissage n'est pas complètement magique et ne permet d'induire une fonction que si on lui fournit suffisamment d'informations en lui spécifiant la *structure* de la fonction qui devra être inférée. En pratique, cela revient à ébaucher un *squelette* de classifieur en omettant certains *paramètres* (des valeurs numériques). L'algorithme d'apprentissage décidera alors à partir du jeu de données quels paramètres optimiseront la capacité de généralisation du programme final.

Les classifieurs linéaires

Pour les tâches de classifications de textes, les classifieurs linéaires ont incontestablement été pendant longtemps la classe de fonctions la plus utilisée, à la fois à cause de leur simplicité et de leurs bonnes performances pour une grande variété de problèmes. Comme tous les classifieurs, ce sont des fonctions qui prennent en entrée le vecteur de caractéristiques représentant une observation et renvoient en sortie une étiquette choisie parmi un ensemble prédéfini. Dans un classifieur linéaire, chaque étiquette possible y est associée à un vecteur de paramètres w^y qui a la même taille que l'observation. Dans notre exemple, il y aura donc trois

vecteurs de paramètres (w^{positif} , $w^{\text{négatif}}$, w^{neutre}) pour un total de $n \times 3$ paramètres.

Chacun de ces paramètres⁸ w_i^y , appelés *poids*, est un nombre réel (positif ou négatif) qui décrira l'importance du i -ème mot pour l'étiquette y (à quel point ce mot *caractérise* l'étiquette) : plus ce poids est grand, plus la présence de ce mot renforce la confiance du classifieur dans l'hypothèse « l'étiquette de x est y » ; à l'inverse, un poids négatif réduira cette confiance. Ainsi, si le mot « génial » est représenté par la 1 703^e caractéristique, on s'attend à ce que $w_{1703}^{\text{positif}}$ soit positif et très grand (indice d'un commentaire favorable) et, à l'inverse, $w_{1703}^{\text{négatif}}$ négatif et très petit.

Une fois tous les poids fixés (nous verrons comment rapidement), un classifieur linéaire déterminera le *score* de chaque étiquette y en calculant le produit scalaire entre la représentation x et le vecteur de paramètres de cette étiquette :

$$\begin{aligned} F_y &= x \cdot w^y \\ &= x_1 \times w_1^y + x_2 \times w_2^y + \dots + x_n \times w_n^y \end{aligned}$$

Le produit scalaire de deux vecteurs x et w^y est un nombre réel qui s'obtient en multipliant les composantes des deux vecteurs de même indice x_i et w_i^y entre elles et en additionnant tous ces produits. Par exemple, le produit scalaire entre les vecteurs (1 2 3) et (4 5 6) est égal à $1 \times 4 + 2 \times 5 + 3 \times 6$, soit 32. Si l'observation est un vecteur binaire, le produit scalaire revient simplement à additionner entre eux tous les poids correspondant aux mots présents dans un document⁹.

Le score F_y est donc une combinaison, dite *linéaire*, des poids des mots présents dans le document, et ceux-ci jouent le rôle d'autant de sources d'information pouvant faire pencher la décision vers une étiquette ou une autre. Comme nous le verrons, les combinaisons linéaires sont au cœur de quasiment tous les modèles et jouent un rôle central dans les réseaux de neurones.

Finalement, le score de l'étiquette y traduit la confiance « combinée » avec laquelle le classifieur associe y à un document. Celle-ci résulte de la fusion de l'ensemble des informations contenues dans la représentation d'un document. Pour prendre une décision et prédire l'étiquette associée à une observation x , il suffit au classifieur de choisir l'étiquette dont le score est le plus grand.

La combinaison des poids apporte ainsi une solution à l'aspect *multi-critères* des décisions décrit plus haut : si un texte contient un mot comme « fantastique » qui exprime un avis très fortement positif et un mot comme « moyen » qui traduit un avis plus mitigé, le poids du premier dans $w^{positif}$ sera très élevé (par exemple 5) et celui du second négatif (par exemple, de -1) ; l'addition de ces deux poids réduira le score de l'étiquette *positif*, mais celui-ci sera toujours élevé et l'étiquette très certainement choisie. Si, par contre, le texte contient, cette fois-ci, trois mots marquant un avis négatif (avec par exemple des scores de -1 , -2 et -3), le score de l'étiquette *positif* sera plus fortement réduit, peut-être suffisamment pour qu'il devienne inférieur à celui de l'étiquette *négatif* et que la décision du classifieur change. Le fait que les poids soient choisis automatiquement à partir des données et ne doivent pas être spécifiés manuellement permet de prendre en compte l'aspect *flou* des critères.

L'algorithme du perceptron

Maintenant que nous avons vu comment un classifieur linéaire réalise ses prédictions, nous pouvons aborder la question essentielle de savoir comment choisir correctement les paramètres de celui-ci afin de garantir que ses décisions soient (le plus souvent) correctes. Cette étape est la seule qui soit, à proprement parler, automatique.

L'algorithme du perceptron, mis au point par Frank Rosenblatt en 1957, est l'une des stratégies d'apprentissage les plus anciennes et

les plus intuitives pour déterminer les paramètres d'un classifieur linéaire à partir d'un ensemble de données annotées, c'est-à-dire des observations et leur étiquettes. Cet algorithme, dit *par correction d'erreurs*, repose sur l'idée d'utiliser le classifieur pour prédire les étiquettes des exemples du corpus d'apprentissage et de *corriger* les poids dès que celui-ci ne donne pas la réponse attendue et fait donc une erreur.

Plus précisément, l'algorithme commencera par initialiser tous les paramètres du classifieur à 0. Puis il choisit un exemple au hasard dans le corpus d'apprentissage et prédit l'étiquette de celui-ci en utilisant la valeur courante de ses paramètres. En comparant sa prédiction à l'étiquette de référence (cette étiquette est connue puisque l'exemple est choisi dans le corpus d'apprentissage), l'algorithme sait si sa réponse est correcte ou non.

Si l'étiquette prédite est la bonne, l'algorithme ne fait rien et recommence la procédure en choisissant un nouvel exemple au hasard. Dans le cas contraire, il corrigera les paramètres du classifieur et augmentera les poids du vecteur de paramètres de l'étiquette de référence (celle qui aurait dû être prédite) en ajoutant à celui-ci le vecteur $\mathbf{x}$. De manière similaire, il pénalisera les paramètres de l'étiquette qui a été prédite par erreur en leur soustrayant $\mathbf{x}$.

Cette opération très simple va renforcer l'association entre les caractéristiques (ici, les mots) de l'exemple courant et l'étiquette de celui-ci : après cette correction, le score de l'étiquette de l'exemple sera plus grand, augmentant ainsi les chances que celle-ci soit choisie si cette observation ou une observation ayant (en partie) des mots (et donc des caractéristiques) en commun doit à nouveau être classée.

Une fois ces mises à jour effectuées, l'algorithme choisira un nouvel exemple au hasard et recommencera les mêmes étapes. En répétant cette procédure (choix aléatoire d'un exemple, classification, mise à jour) un très grand nombre de fois (il existe

des méthodes pour déterminer quand arrêter l'apprentissage), l'algorithme trouvera progressivement des poids qui associeront chaque exemple à son étiquette de référence.

Grâce à cette méthode itérative, l'algorithme fera graduellement émerger quantité de *corrélations* ou d'*associations* entre étiquettes et caractéristiques bien souvent invisibles à l'œil nu : chaque fois qu'un mot apparaît dans un commentaire exprimant un avis positif mal classé, son importance pour l'étiquette *positif* sera renforcée ; si jamais cette importance devient trop grande et que le classifieur commence à reconnaître tous les commentaires contenant ce mot comme positif même en présence d'un grand nombre de mots connotés négativement, la deuxième règle de mise à jour entrera en jeu et diminuera ce poids.

La force des classifieurs linéaires est d'arriver à capturer ce type d'associations pour toutes les caractéristiques, aussi bien celles décrivant des mots marquant clairement une opinion (comme « génial ») sur lesquels auraient naturellement porté des règles, que celles dont le lien avec la tâche n'a *a priori* rien d'évident. Contrairement aux systèmes experts du chapitre 1 (voir [ici](#)), il n'y a plus besoin, pour prendre la décision, de spécifier explicitement la règle à suivre, ni même d'identifier les mots devant être considérés. Il suffit de fournir à l'algorithme un grand nombre d'exemples avec leur étiquette et surtout une représentation adaptée (même si ce dernier élément est souvent oublié) pour que celui-ci repère l'ensemble des régularités qui lui permettront de prendre des décisions optimales (dans la plupart des cas).

Une théorie pour l'apprentissage statistique

L'algorithme du perceptron, dont nous venons d'expliquer le fonctionnement, a marqué un tournant important dans l'histoire de l'IA et, plus généralement, de l'informatique. Inventé par Frank Rosenblatt, un psychologue qui cherchait à modéliser le fonctionnement du cerveau humain (le perceptron est souvent,

aujourd'hui encore, présenté comme le premier « neurone artificiel » et l'ancêtre des réseaux de neurones), cet algorithme a suscité de nombreux espoirs sur la possibilité que les ordinateurs acquièrent des capacités humaines comme la vision, la compréhension des langues ou le raisonnement. Son inventeur déclara même que le perceptron était « la première machine capable d'avoir une idée originale »¹⁰, et plusieurs articles dans des journaux grand public firent des comptes rendus exaltés de son invention qui ne sont pas sans rappeler l'enthousiasme suscité par la présentation de ChatGPT : le *New Yorker* présenta même le perceptron comme le premier rival à l'intelligence humaine¹¹.

Le perceptron a, en effet, prouvé à une époque où l'informatique était encore balbutiante (le premier perceptron a été *programmé* par Rosenblatt sur un IBM 704, un ordinateur de 5 tonnes qui occupait une pièce entière) qu'il était possible qu'un ordinateur « apprenne en faisant » plutôt qu'en étant programmé explicitement pour réaliser une tâche. Pourtant, très vite, de nombreux travaux ont identifié plusieurs limites des perceptrons, soulignant notamment leur pouvoir d'expression limité.

En effet, comme leur nom l'indique, les classifieurs linéaires ne peuvent *réagir* que de manière linéaire à une variation de leur entrée : si une caractéristique est deux fois plus grande dans une observation que dans une autre, son importance ne pourra être que deux fois plus grande. Ils sont donc incapables de prendre des décisions reposant sur la détection de seuils¹² (par exemple pour capturer l'information qu'un mot dépasse une certaine fréquence) ou la combinaison de caractéristiques (comme identifier que deux mots sont présents simultanément dans un texte afin, par exemple, de capturer qu'un adjectif comme « génial » apparaît en même temps que « pas »).

Ces critiques ont été synthétisées dans *Perceptrons*, un célèbre livre publié en 1969 par Seymour Papert et Marvin Minsky, l'un des papes de l'IA et ardent défenseur des méthodes symboliques,

dont les travaux dans le domaine furent récompensés par le prix Turing l'année suivante. L'ouvrage a marqué un tournant dans l'histoire de ce champ alors naissant : il est, de l'avis général, la cause principale de l'abandon relatif des méthodes statistiques en IA pendant de longues années au profit de stratégies fondées sur la logique. La publication de ce livre joua également un rôle certain dans ce que les historiens du domaine ont appelé le « premier hiver », une période allant de 1974 à 1980 et qui fut marquée par une chute des financements et de l'effort de recherche.

L'histoire du perceptron ne s'arrêta cependant pas là. La publication de *Perceptrons* donna effectivement un coup d'arrêt aux recherches sur les méthodes statistiques en IA, mais il ne faut pas oublier que, si le rôle de Rosenblatt pour populariser celles-ci auprès des chercheurs de ce domaine est indéniable, des méthodes similaires étaient étudiées dès les années 1930 par des mathématiciens et des statisticiens, comme Ronald Fisher, le père de la statistique moderne, qui avait introduit une méthode très proche : l'analyse discriminante linéaire.

Les recherches sur le perceptron ont continué dans ces communautés et ont donné naissance à un nouveau domaine, l'apprentissage statistique, qui a progressivement construit un cadre théorique complet réussissant à formaliser le problème de l'apprentissage (comment inférer une fonction à partir d'un ensemble fini d'exemples), des garanties théoriques fortes sur la capacité d'un classifieur à généraliser, et de nombreuses méthodes toujours utilisées aujourd'hui. Il fallut cependant attendre la fin des années 1990 pour que des approches statistiques, issues notamment de ces travaux, soient utilisées en TAL.

De manière assez remarquable, le perceptron illustre deux éléments essentiels au cœur de la théorie de l'apprentissage statistique et de toutes les méthodes d'apprentissage, y compris celles employées aujourd'hui par tous les giga-modèles.

Comprendre ces éléments dans le cas beaucoup plus simple du perceptron, nous permettra de mieux appréhender les limites des réseaux de neurones que nous introduirons au prochain chapitre.

Le premier de ces éléments est le principe de la minimisation du risque empirique (ERM pour *empirical risk minimisation*) qui stipule qu'il faut fixer les paramètres d'un classifieur de manière à minimiser le nombre d'erreurs sur l'ensemble d'apprentissage (une quantité appelée *risque empirique* dans la théorie de l'apprentissage statistique). C'est exactement ce que fait le perceptron ! Ce principe définit un critère qui permet de choisir les paramètres de n'importe quel classifieur et formalise l'apprentissage comme un problème d'optimisation, une branche des mathématiques dont l'objectif est de construire des méthodes sachant trouver efficacement la plus petite ou la plus grande valeur d'une fonction.

Si le principe ERM apparaît très naturel, il a fallu attendre les années 1990 pour que deux mathématiciens russes, Vladimir Vapnik et Alexey Chervonenkis, lui apportent une justification théorique. Ces travaux ont donné naissance à l'une des principales théories de l'apprentissage statistique, la théorie VC (selon les initiales de ses deux concepteurs), qui explique, entre autres¹³, sous quelles conditions un classifieur dont les paramètres ont été choisis selon le principe ERM est capable de généraliser, c'est-à-dire de prédire correctement les étiquettes associées à des entrées qui n'apparaissent pas dans le corpus d'apprentissage.

Le principal résultat de Vapnik et Chervonenkis prouve que, avec une forte probabilité¹⁴, les paramètres d'un classifieur choisi selon le principe ERM sont *optimaux*, dans le sens où ils minimisent le taux d'erreur sur n'importe quel autre ensemble de données. Employé sur de *nouvelles* observations, ce classifieur prendra moins de mauvaises décisions que tout autre classifieur. Ce résultat n'est toutefois valable que si les données du corpus d'apprentissage (celles qui ont servi à choisir les paramètres) sont

similaires à celles sur lesquelles il sera utilisé. Cette condition se traduit mathématiquement par la notion de données *identiquement distribuées* et constitue, comme nous le verrons bientôt, l'une des principales limites de l'apprentissage statistique.

La manière dont le perceptron minimise l'erreur est un deuxième élément caractéristique du fonctionnement des méthodes à base d'apprentissage. Celle-ci est en effet en tout point similaire à la manière dont la quasi-totalité des modèles sont entraînés aujourd'hui encore. En effet, l'algorithme du perceptron peut s'interpréter comme une *descente de gradient stochastique*, une méthode d'optimisation classique pour trouver (sous certaines conditions) le minimum d'une fonction arbitraire.

Qu'entend-on par descente de gradient stochastique ? Il s'agit d'une technique d'optimisation itérative qui repose sur la répétition de deux étapes en tout point similaires à celles qui se trouvent au cœur de l'algorithme du perceptron, mais qui peuvent être généralisées à tout type de classifieur. La première consiste à choisir un exemple au hasard (c'est l'aspect « stochastique » de la méthode d'optimisation) et à comparer l'étiquette que le classifieur prédit pour cet exemple à la sortie attendue. Cette comparaison permet de calculer l'*erreur* du classifieur : dans le cas du perceptron, cette erreur vaut soit 1 (quand les deux étiquettes diffèrent), soit 0 (quand elles sont identiques), mais il est tout à fait possible pour d'autres tâches que l'erreur prenne une valeur continue.

La seconde étape consiste alors à modifier les paramètres du classifieur de manière à réduire cette erreur. L'astuce consiste à exprimer l'erreur comme une fonction des paramètres. Grâce à des règles de calcul très simples, on calcule alors le *gradient* de cette fonction (sa variation, en termes moins mathématiques) afin de savoir où elle se minimise.^{[15](#)}

Comme dans le cas du perceptron où, rappelons-le, la mise à jour consiste à ajouter ou à soustraire le vecteur d'observation $\mathbf{x}$ aux

paramètres, ce gradient dépend toujours de l'observation. Ainsi, aussi bien pour les classifieurs linéaires, que pour les réseaux de neurones les plus complexes, les paramètres sont choisis de manière à ce que le modèle reproduise les associations entre entrées et sorties du corpus de données. L'apprentissage va alors renforcer les associations entre les caractéristiques pertinentes de l'observation et l'étiquette, de manière à faire émerger des *régularités* : par exemple, à force d'augmenter le poids du mot « génial » dans le vecteur de la classe *positif*, l'algorithme d'apprentissage arrivera à détecter que ce mot est un indice fort d'un commentaire positif et pourra exploiter cette information pour faire de nouvelles prédictions. Il n'est pas question ici d'innovation ou de créativité, mais uniquement de reproduction et d'imitation.

S'il est facile de comprendre comment la descente de gradient stochastique permet aux modèles linéaires de découvrir des corrélations utiles pour prédire les étiquettes de nouveaux exemples, il est particulièrement surprenant que des règles de mise à jour aussi simples soient au cœur de tous les progrès récents du TAL et de l'IA observés ces dernières années. Ce sont pourtant exactement les mêmes principes et (quasiment) les mêmes équations qui sont utilisés pour apprendre les paramètres des giga-modèles leur permettant de parler, de répondre aux questions, de dialoguer, voire, comme nous le verrons bientôt, d'apprendre à construire des représentations du monde. Comprendre comment une méthode aussi simple permet de faire émerger toute cette *intelligence* est, à l'heure actuelle, une question scientifique majeure.

Limites des classifieurs linéaires

Les classifieurs linéaires répondent à de nombreux besoins et servent, aujourd'hui encore, à résoudre beaucoup de problèmes de classification qui, rappelons-le, consistent à associer une

étiquette à un texte, celle-ci pouvant permettre d'identifier les spams, l'urgence d'un mail, la polarité d'un commentaire, le type de problème décrit dans un rapport d'incident, etc. Ces classifieurs constituent également une brique essentielle pour traiter des problèmes plus complexes, et notamment ceux reposant sur la génération de textes comme la traduction, le résumé automatique ou les systèmes de dialogue (chatbot).

Les problèmes de génération peuvent, en effet, se modéliser comme une suite de décisions et donc de problèmes de classification. Par exemple, il est possible de traduire un texte en utilisant un classifieur qui choisit, à partir d'un vecteur de caractéristiques décrivant la phrase source et le début de la traduction déjà générée (le *préfixe* de la traduction), le prochain mot à générer parmi tous les mots possibles de la langue cible, ce processus étant répété jusqu'à la prédiction de la fin de la phrase. Dans cette approche, la traduction est réduite à une suite de problèmes de classification en tout point semblable à la prédiction de la polarité d'un texte. La seule différence (et ce n'est pas un détail !) est le nombre d'étiquettes considérées (le nombre de *réponses* possibles) : trois dans le système décrit au début de ce chapitre contre plusieurs dizaines de milliers (une pour chaque mot du vocabulaire) dans un système de traduction.

À partir des années 2000, les méthodes à base d'apprentissage sont devenues la solution de choix pour traiter tous les problèmes de TAL, même si plusieurs limites ont été progressivement identifiées. Une première insuffisance des modèles linéaires (et de tous les autres classifieurs) découle directement des garanties théoriques de l'apprentissage statistique, et plus précisément de la nécessité de disposer de données d'apprentissage similaires à celles sur lesquelles le classifieur sera utilisé. Ainsi un classifieur appris uniquement sur des commentaires de livres donnera de très bons résultats s'il est employé exclusivement dans ce cadre spécifique ; il est très probable que ses performances se dégradent sur des critiques de films, et même qu'elles deviennent

catastrophiques sur des documents financiers (par exemple ceux donnant l'avis d'experts sur la santé financière d'une entreprise).

En effet, ces différents textes appartiennent à des *domaines* différents : s'ils sont tous écrits dans une même langue, le vocabulaire employé et la construction des phrases vont fortement varier – il est peu probable que des critiques de livres parlent d'une « réalisatrice », de « plan » ou même de « remake », et les commentaires publiés par des utilisateurs contiennent, par exemple, très certainement des tournures moins soutenues et plus d'erreurs que ceux publiés par un critique professionnel. Il est donc plus que vraisemblable que les poids associés à certains mots soient mal estimés lors de l'apprentissage, voire que les régularités découvertes lors de l'apprentissage sur un domaine ne soient plus forcément pertinentes pour un autre domaine.

Mais comment savoir si les données sur lesquelles on utilise un classifieur sont issues du même domaine que celui sur lequel il a été appris ? Il est difficile de répondre, tant la notion de domaine (traduction *pratique* de la propriété de données « identiquement distribuées ») couvre un vaste éventail de variations possibles : variation de thèmes (la manière dont on exprime son opinion sur un livre n'est pas la même que celle que l'on utilise pour donner son avis sur les performances d'une équipe de football), de média (le français utilisé sur les réseaux sociaux ne ressemble pas nécessairement à celui qui est utilisé dans un article du journal *Le Monde*), de l'âge du rédacteur (est-ce qu'un adolescent utilise les mêmes expressions qu'une personne plus âgée ?), etc. Toutes ces variations peuvent (ou non) avoir un impact sur les performances du classifieur et, en pratique, la notion de domaine est souvent définie uniquement de manière expérimentale lorsque ces performances baissent ou ne sont pas aussi bonnes qu'attendues.

La nécessité d'entraîner les classifieurs sur des données du même domaine que celui sur lequel il sera utilisé a pendant longtemps été un frein important à l'utilisation des méthodes

d'apprentissage. En effet, l'effort de collecte et d'annotation de grands corpus de données est, comme expliqué dans l'encadré p. 66, très important tant par le temps nécessaire à l'annotation que par la pénibilité de ce travail et, finalement, à son coût. Ce besoin d'informations étiquetées est exacerbé par une seconde limite des classifieurs linéaires : l'indépendance des caractéristiques. Dans ce type de classifieur, le rôle que chaque caractéristique joue dans la décision (son poids) est déterminé à partir des données d'apprentissage sans prendre en compte la valeur des autres caractéristiques.

Cela est particulièrement problématique dans les cas, comme celui décrit dans ce chapitre (voir [ici](#)), d'une représentation basée uniquement sur les mots et donc sur la détection d'une suite de caractères (ce que l'on appelle une *forme de surface*) sans qu'aucun *sens* ne soit rattaché à celle-ci. En effet, toute variation, même minime, du texte représenté résultera dans une caractéristique complètement différente et décrite par un poids qui lui sera propre. Ainsi, des mots intuitivement proches comme « génial » et « brillant » (des synonymes dans la plupart des dictionnaires), mais aussi des termes comme « géniaux », « géniale » ou « génialement » (des formes dérivées de « génial ») seront tous représentés par des caractéristiques différentes, et les poids correspondants n'auront aucun lien entre eux : même si le classifieur découvre que « génial » est un mot important pour caractériser les commentaires positifs, il ne pourra *partager* cette information avec les caractéristiques des mots « brillant » ou « géniale », et les poids afférents devront être déterminés *de zéro*, sans pouvoir bénéficier des informations déjà extraites. À cause de cette incapacité à partager des informations entre caractéristiques, on doit disposer de corpus suffisamment grands pour observer tous les mots, mais également toutes leurs formes.

En plus de ces difficultés *pratiques* qui peuvent être résolues au prix de nombreux efforts (par exemple en constituant de grands corpus de données annotées ou en concevant des programmes

capables d'identifier les mots d'une même famille et de les regrouper sous une même forme), on peut se demander si ces méthodes d'apprentissage permettent de développer des systèmes réellement *intelligents* : le perceptron ne fait, finalement, que découvrir des corrélations entre étiquettes et caractéristiques. Les décisions qu'il prend reposent uniquement sur sa capacité à détecter et à exploiter des combinaisons de ce que l'on appelle souvent des *signaux faibles* difficilement identifiables même par des experts du domaine. En définitive, leur *intelligence*, ou du moins, pour être plus précis, leur capacité à prendre de bonnes décisions, est intimement liée à la qualité des représentations choisies par le concepteur du programme et qui lui sont données en entrée.

Ces représentations ont, en effet, deux propriétés essentielles qui n'ont pas encore été assez soulignées : les caractéristiques sont définies *manuellement* et *arbitrairement*. C'est le programmeur qui choisit, sans aucune contrainte, les informations qui lui paraissent pertinentes pour l'application visée en fonction de sa connaissance du problème ou des informations fournies par un spécialiste du domaine. Ces caractéristiques peuvent ainsi être très simples (liste de tous les mots contenus dans le document), mais aussi reposer sur des connaissances extérieures comme des dictionnaires ou des listes de mots pré-identifiés, et même résulter d'un traitement complexe fondé, par exemple, sur des requêtes sur l'Internet ou sur les prédictions d'un autre classifieur. Une limite importante des classifieurs linéaires est de reposer sur une représentation fixe des observations : si celle-ci contient les informations pertinentes (ou au moins une partie de celle-ci), les performances seront bonnes, voire excellentes ; mais ces classifieurs ne parviendront jamais à prendre de bonnes décisions si les représentations choisies par leurs concepteurs ne sont pas adaptées.

Par exemple, si l'on considère la tâche de stylométrie¹⁶ qui consiste à déterminer qui, parmi un ensemble prédéfini d'auteurs,

a écrit un texte donné, il est peu probable qu'un modèle linéaire soit capable de distinguer les œuvres de Molière de celles de Racine en considérant simplement la liste des mots apparaissant dans une pièce (ce que fait la représentation en sac de mots habituellement utilisée). En revanche, en utilisant une représentation plus riche basée sur des connaissances précises du domaine (par exemple, le fait que Molière préfère les phrases courtes ou que Racine utilise plus de tournures passives et respecte souvent la règle des trois unités du théâtre classique), un classifieur linéaire peut tout à fait réussir à distinguer parfaitement ces deux auteurs. Il est toutefois légitime de se demander si un programme à qui on a soufflé qu'il fallait comparer le nombre de phrases au passif pour prendre sa décision est réellement intelligent.

Plus que la volonté de développer des systèmes *intelligents*, ce sont des questions très pratiques (à savoir, principalement l'indépendance des caractéristiques et les besoins accrus en données annotées que cette limite implique) qui ont poussé, progressivement, les chercheurs en TAL vers une autre classe de fonction, les réseaux de neurones, dont nous allons maintenant voir les grands principes.

Annoter des données

La mise en œuvre de méthodes d'apprentissage statistique nécessite de (grands) corpus d'exemples annotés (l'apprentissage est supervisé !). Il existe aujourd'hui deux grandes méthodes pour constituer ces corpus. La première méthode consiste à collecter des données et à recruter des annotateurs dont le travail consistera spécifiquement à remplacer le classifieur en donnant, pour chaque exemple, la réponse que celui-ci devra prédire. Cette annotation manuelle est une opération longue, coûteuse, répétitive et ingrate. Par exemple, la constitution du *Prague Dependency Treebank*, un corpus arboré (*treebank*) contenant 1,8 million de mots en tchèque annotés morphologiquement et syntaxiquement¹⁷, a nécessité le travail de 22 personnes (jusqu'à 17 simultanément) pendant

5 ans avec un coût total estimé à 600 000 dollars¹⁸. Cet *effort d'annotation* est également conséquent pour la création de corpus de transcription (la transcription est la tâche qui consiste à *traduire* un fichier audio en texte) : on juge qu'il faut généralement entre 20 et 60 heures de travail pour transcrire manuellement 1 heure d'enregistrement audio.

Il existe aujourd'hui de nombreuses plateformes de micro-travail ou *travail au clic* permettant de recruter facilement des personnes pour annoter les corpus, que ce soit en TAL ou pour tous les autres domaines de l'IA : Microworkers, Ferpection, Werk et, la plus célèbre d'entre elles, Amazon Mechanical Turk, lancée par Amazon en 2005. Évaluer des traductions, prononcer des mots, transcrire des manuscrits, étiqueter des images, etc. Ce sont ces petites mains de l'ombre qui, dans des conditions de travail (horaires, salaire, etc.) que de nombreux articles et rapports officiels jugent bien souvent « indignes »¹⁹, produisent les données permettant à la machine d'apprendre et de progresser aussi vite.

Ces tâches titanesques d'annotations reposent sur le travail d'une armée de travailleurs : en 2019, une étude de trois chercheurs, Antonio Casilli, Paola Tubaro et Clément Le Luëc, recensait 266 000 micro-travailleuses ou micro-travailleurs occasionnels et 15 000 très actifs en France et évaluait leur nombre à 90 millions dans le monde²⁰. Pourtant, ce travail est bien souvent invisibilisé, donnant une impression d'algorithmes intelligents et autonomes. Il ne faut toutefois pas oublier que, pour chaque succès et chaque avancé de l'IA, il y a très souvent un humain derrière la machine, et que les bonnes performances des méthodes actuelles ne seraient pas possibles sans le travail, en coulisse, de milliers de travailleurs précaires. La question des conditions dans lesquelles les données nécessaires aux algorithmes d'apprentissage sont annotées continue d'être problématique : comme l'a révélé une enquête du *Time*²¹, les prouesses de ChatGPT n'ont été rendues possibles que par le travail de plus de 1 000 annotateurs, recrutés par un sous-traitant d'OpenAI au Kenya et payés moins de 2 dollars de l'heure.

Une seconde approche pour collecter des données à moindre coût consiste à utiliser les giga-octets de données produits par les utilisateurs lors de leurs interactions virtuelles sur le web ou dans des applications. Ainsi, chaque fois qu'une personne donne son avis sur un film (que ce soit par un commentaire ou par une note), corrige une traduction de DeepL ou Google Translate, clique sur un lien ou retweete un message, elle fournit des informations précieuses qui, une fois agrégées, nettoyées

et analysées, iront enrichir les corpus des entreprises fournissant gracieusement ces services.

Le besoin de collecter le plus de données possibles justifie (en partie) l'intérêt pour une entreprise de laisser ses produits librement accessibles : ce n'est pas par altruisme que OpenAI ne facture pas encore l'utilisation de ChatGPT, mais parce que cela lui permet de récupérer gratuitement ou presque des exemples de questions, des évaluations des réponses fournies par ce service, et tout un tas d'informations qui lui permettront d'enrichir ChatGPT et de le rendre encore plus performant. Bien plus que le code ou les idées, le contrôle des données est aujourd'hui l'une des principales sources de valeur des entreprises du numérique.

Ce n'est donc pas un hasard si, après avoir accusé OpenAI d'utilisation abusive de ses données²², X (anciennement Twitter) a annoncé vouloir limiter le nombre de tweets pouvant être lus par un utilisateur à 10 000 chaque jour²³. Les raisons données par Elon Musk, le PDG de X, sont très claires : « Quasiment toutes les boîtes qui font de l'IA, des start-up aux plus grands groupes du monde, ramassaient de grandes quantités de données [...]. C'est un peu frustrant de devoir ajouter, en urgence, un nombre important de serveurs pour le seul fait de justifier la valorisation indécente de certaines start-up de l'IA. ». C'est aussi et surtout un moyen de garantir que ses données ne servent à entraîner que ses propres IA et non celles de ses concurrents, un nouvel épisode dans la guerre que se livrent les géants du web autour du développement des giga-modèles.

Sans disposer nécessairement d'une application avec des millions d'utilisateurs, il est possible de récupérer très facilement sur l'Internet d'impressionnantes quantités de données pouvant servir pour l'apprentissage de système. Ainsi, on trouve sur OpenSubtitles²⁴ les sous-titres de près de 7 millions de films. Avec un peu d'ingéniosité²⁵, il est possible d'en extraire plus de 45,2 millions de phrases en français traduites en anglais. Au total, ce sont plus de 850 millions de phrases traduites entre le français et l'anglais qui peuvent être trouvées sur le web²⁶. À titre de comparaison, les 1 664 pages (dans l'édition poche) du roman de James Joyce *Ulysse*, ne contiennent « que » 43 000 phrases.

L'utilisation de ces masses de données non contrôlées pour entraîner des programmes de TAL pose de nombreux problèmes : violation des droits d'auteur, présence de biais (d'âge, de genre, etc.), textes à caractères injurieux ou raciste, etc. Avec l'utilisation croissante des outils de TAL

dans notre quotidien, il devient de plus en plus urgent d'apporter des solutions à tous ces problèmes (voir chapitre 5, [ici](#)).

3

APPRENDRE À REPRÉSENTER LE MONDE

Les réseaux de neurones sont une classe de fonctions capables de découvrir automatiquement une représentation « optimale » des entrées du classifieur. Introduits dès les débuts de l'informatique, ils ont attendu les années 2010 pour atteindre une forme de maturité et être manipulés correctement par les informaticiens à grande échelle. Aujourd'hui, les transformeurs, une architecture neuronale introduite par Alphabet en 2017, permettent de découvrir automatiquement des représentations du texte qui capturent de nombreuses informations sur la syntaxe et la sémantique des phrases. Cette architecture a ouvert les portes à une nouvelle génération de systèmes de TAL aux performances surprenantes.

Il est difficile d'imaginer une technologie plus sous les feux des projecteurs que les réseaux de neurones et l'apprentissage profond (*deep learning*), deux termes qui, nous le verrons bientôt, désignent exactement la même chose. Si ceux-ci sont effectivement, sans aucun doute, la pierre angulaire des récents progrès des systèmes de TAL, et plus généralement de toute l'IA, les réseaux de neurones sont une technologie ancienne, presque aussi vieille que l'informatique, dont les tenants et les aboutissants sont connus depuis de nombreuses années.

Du point de vue de l'apprentissage statistique, les réseaux de neurones sont *juste* une classe de fonctions qui apportent une solution aux limites des modèles linéaires évoquées à la fin du chapitre précédent, en permettant notamment de déterminer automatiquement des représentations des entrées que le classifieur doit traiter.

Malgré leur nom et les fantasmes qui leur sont associés, nul besoin pour comprendre ce nouveau modèle de se plonger dans un livre de neurosciences ! La métaphore biologique, si elle est particulièrement attrayante et intrigante, n'a plus qu'un rôle historique et, pour comprendre les réseaux de neurones, il est plus utile d'avoir quelques notions d'algèbre : les réseaux de neurones peuvent se définir très facilement à partir de quelques opérations sur des matrices.

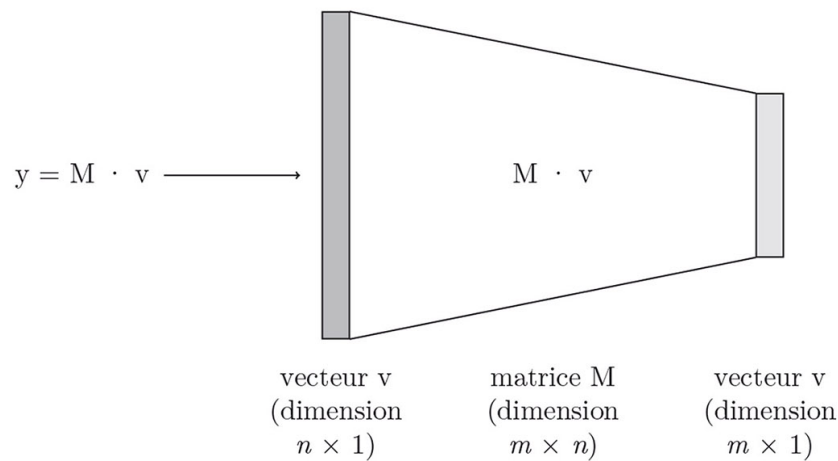
Les matrices

Une matrice peut être vue comme une feuille de calcul dans un tableur : c'est un ensemble de nombres organisés en lignes et en colonnes. Les dimensions d'une matrice correspondent à son nombre de lignes et de colonnes (toujours donné dans cet ordre). Voici, par exemple, une matrice 3×5 :

$$M = \begin{pmatrix} 1 & 4 & 7 & 10 & 13 \\ 2 & 4 & 8 & 11 & 14 \\ 3 & 6 & 9 & 12 & 15 \end{pmatrix}$$

Les vecteurs que nous avons manipulés jusqu'à présent peuvent être vus comme des matrices $n \times 1$ pour les vecteurs colonnes ou des matrices $1 \times n$ pour des vecteurs lignes. Il est également possible de définir des vecteurs en extrayant d'une matrice tous les nombres d'une même ligne ou d'une même colonne.

La multiplication¹ entre une matrice M (de dimension $m \times n$) et un vecteur colonne de dimension $n \times 1$ est au cœur de la définition des réseaux de neurones. Le résultat de cette opération est un vecteur de m éléments, dont le i -ème élément est défini comme le produit scalaire entre v et la i -ème colonne de la matrice M . Par la suite, nous représentons schématiquement cette opération de la manière suivante :



Ce schéma permet de mettre en évidence que la multiplication de v par M permet de *transformer* le vecteur en un nouveau vecteur de taille différente. Les composantes de ce nouveau vecteur sont définies comme des combinaisons linéaires des composantes du vecteur v .

L'utilisation de matrice permet de simplifier (au moins pour l'informaticien) la description du classifieur linéaire que nous avons vue au chapitre précédent : en *empilant* côte à côte les vecteurs de paramètres de chaque étiquette, on peut construire une *matrice de paramètres* de dimension $n \times k$ où n est la taille du vecteur d'observations et k le nombre d'étiquettes. En multipliant cette matrice par le vecteur d'observations (le nombre de caractéristiques), on obtient un vecteur de dimension k dont la i -ème composante correspond au produit scalaire entre v et la i -ème colonne de M , et donc au score de la i -ème classe. Comme nous allons le voir, cette formulation va servir de base à la définition des réseaux de neurones.

Apprendre à représenter : les réseaux de neurones

Une nouvelle classe de fonctions

Les réseaux de neurones sont une classe de fonctions particulière, utilisée pour résoudre des problèmes de classification, tout comme les classifieurs linéaires. Leur principale force réside dans leur capacité à découvrir automatiquement de meilleures représentations des entrées. Pour cela, un réseau de neurones ne part pas de rien : il va apprendre à *transformer* la représentation

initiale fournie par le programmeur pour construire une *représentation cachée*. Cette nouvelle représentation étant déterminée automatiquement à partir du corpus d'apprentissage, on peut espérer qu'elle sera mieux *adaptée* au problème et qu'elle permettra au classifieur de prendre de meilleures décisions.

La construction de la représentation cachée se déroule en deux étapes. La première généralise la formulation matricielle d'un classifieur linéaire (voir encadré p. 72) en y introduisant une deuxième matrice de paramètres. Les valeurs de cette matrice sont là aussi déterminées à partir des données d'apprentissage. En multipliant le vecteur représentant l'observation (de taille $n \times 1$) par cette matrice de taille $h \times n$, que nous noterons $\mathbf{M}^{proj}$, nous obtenons h nouvelles caractéristiques représentant une observation donnée. Chacune de ces caractéristiques sera une combinaison linéaire des caractéristiques initiales.

La représentation cachée est ensuite obtenue en appliquant une fonction non linéaire à chaque composante de ce vecteur. Cette fonction permet de « casser » la relation linéaire entre la valeur d'une caractéristique et le score d'une étiquette, et d'éviter que le résultat d'une combinaison linéaire ne devienne trop petit ou trop grand. Une fonction très couramment utilisée aujourd'hui est la fonction ReLU², qui agit comme un filtre et annule les caractéristiques négatives en sélectionnant uniquement les caractéristiques positives.

Intuitivement, ces deux étapes permettent de construire tout un éventail de caractéristiques *complexes* en combinant et en filtrant les informations contenues dans la représentation initiale. Il est, par exemple, possible de détecter si deux mots sont présents simultanément dans la même phrase³ ou de déterminer si une caractéristique (ou, pour être plus précis, une combinaison de caractéristiques) dépasse un seuil qui sera déterminé à partir du corpus d'apprentissage.

Une fois la représentation cachée construite, une deuxième matrice, M^{score} de taille $k \times h$, va être utilisée pour générer un vecteur de k scores à partir de celle-ci, exactement comme dans un classifieur linéaire. Une fois ces scores calculés, le réseau n'aura qu'à choisir l'étiquette correspondant au plus grand score pour faire sa prédiction. La figure ci-dessous donne un aperçu général de l'architecture de ce premier réseau de neurones et montre clairement que celui-ci est une généralisation directe d'un classifieur linéaire :

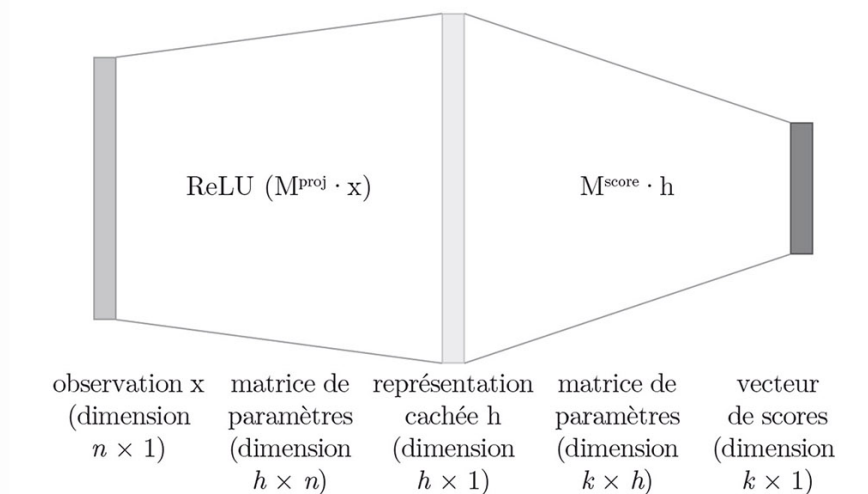


Figure 6. Architecture du réseau de neurones le plus simple. Ce réseau à propagation avant à une couche transforme le vecteur d'entrée (représenté en gris foncé) successivement en une représentation cachée et un vecteur de score.

Un réseau de neurones, tout comme un classifieur linéaire, utilise une combinaison linéaire de caractéristiques pour prendre une décision. Cependant, au lieu d'apprendre uniquement à combiner les caractéristiques préalablement choisies par un expert, il a la capacité de découvrir, à partir des données, une transformation de ces informations lui permettant d'enrichir la représentation. Cette nouvelle capacité s'accompagne toutefois d'une perte d'interprétabilité des représentations. Contrairement aux représentations manipulées par les classifieurs linéaires, dont la sémantique est clairement définie (c'est le programmeur qui a choisi chaque caractéristique en fonction de sa connaissance du problème), la représentation construite automatiquement par le

réseau de neurones est, comme son nom l'indique, cachée. En pratique, il est impossible de savoir ce que décrit une composante de cette représentation, car elle repose, même dans les cas les plus simples, sur une combinaison (qui est, en plus, transformée par une fonction non linéaire) de toutes les caractéristiques qui lui sont données en entrée.

L'architecture que nous venons de voir, dite à *propagation avant* (*feed-forward*) définit un classifieur très puissant : plusieurs études théoriques ont démontré⁴, dès 1989, qu'elle permet de construire un *approximateur universel* capable, si l'on choisit correctement la dimension de la couche cachée et la fonction non linéaire, de représenter n'importe quelle fonction. En pratique, on préfère souvent considérer un réseau de neurones composé de plusieurs *couches* et non pas d'une seule. Ce choix permet généralement d'obtenir une meilleure performance en généralisation à partir d'un ensemble d'apprentissage contenant moins d'exemples.

Contrairement au réseau monocouche que nous venons de décrire, qui ne construit qu'une seule représentation cachée, un réseau multicouche en construit plusieurs successivement. Chaque couche, définie par une matrice de paramètres et une fonction non linéaire, permet de transformer la représentation construite par la couche précédente. La figure ci-dessous donne un exemple d'architecture comportant trois couches et construisant donc trois représentations cachées.

Pour réaliser une prédiction, la représentation vectorielle initiale sera successivement multipliée par les matrices de paramètres de chaque couche, créant ainsi une série de représentations cachées intermédiaires et, finalement, un vecteur de scores. Cette *traversée* du réseau au cours de laquelle l'entrée est progressivement transformée, de proche en proche, en sortie, est appelée une *phase de propagation avant* (phase *forward*).

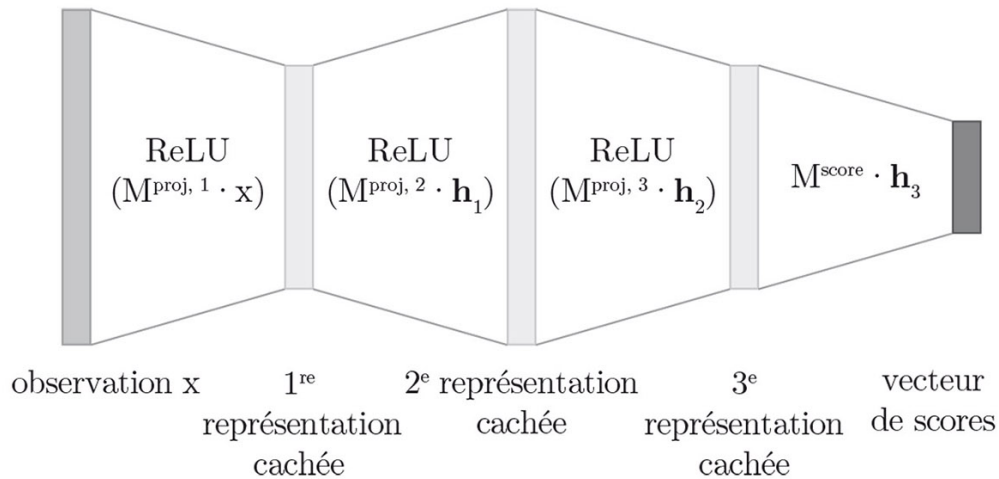


Figure 7. Exemple de réseau à propagation avant avec trois couches cachées. Le vecteur d'entrée traverse le réseau de gauche à droite, subissant successivement trois transformations, chaque transformation étant obtenue par une multiplication matricielle suivie de l'application d'une fonction non linéaire. La dernière transformation produit la représentation cachée *finale* à partir de laquelle un vecteur de scores est prédit.

Intuitivement, l'empilement de couches simplifie la découverte de relations complexes entre les caractéristiques de l'observation en décomposant la construction de la représentation cachée. La succession des couches permet, en effet, de créer une représentation *hiérarchique* des observations, chaque couche apprenant à extraire des caractéristiques à différents niveaux d'abstraction. Les premières couches capturent des caractéristiques de base en combinant les informations de la représentation initiale de l'entrée, tandis que les suivantes combinent ces caractéristiques grossières pour en former des variantes plus complexes qui reposent sur des combinaisons de combinaisons des caractéristiques initiales.

Bien que ces caractéristiques complexes pourraient être apprises directement à partir de la représentation initiale, il est généralement plus efficace de les construire de manière progressive. L'utilisation, à partir des années 2000, de réseaux de plus en plus profonds, c'est-à-dire composés de plus en plus de couches, permettant d'apprendre des représentations de plus en

plus complexes, a poussé les chercheurs à parler d'*apprentissage profond* (*deep learning*) pour bien marquer la différence avec les premiers réseaux de neurones.

Apprentissage d'un réseau de neurones

La définition des fonctions non linéaires, du nombre de couches cachées et de leur taille (les couches n'ont pas nécessairement la même taille ni la même fonction non linéaire) détermine l'*architecture* du réseau de neurones. Le choix de l'architecture est essentiel : c'est, en effet, elle qui déterminera la capacité de généralisation du réseau de neurones. Si l'un de ces paramètres est mal choisi, le réseau ne saura prédire les étiquettes des nouveaux exemples. Il n'existe pas, à l'heure actuelle, de théorie pour guider ces choix, et la seule méthode consiste à expérimenter de nombreuses combinaisons possibles pour déterminer celle qui offre les meilleures performances.

Une fois l'architecture du réseau de neurones fixée, il est possible d'entraîner l'ensemble de ses paramètres : la matrice qui calcule les scores, et surtout les matrices qui créent les représentations cachées. Le processus d'apprentissage d'un réseau de neurones repose sur une descente de gradient stochastique et est en tout point identique à celui d'un classifieur linéaire. Le modèle considère chaque exemple du corpus d'apprentissage un par un, compare la prédiction qu'il génère à l'étiquette de référence, et s'il commet une erreur, ajuste tous les paramètres pour corriger sa prédiction et reproduire la sortie souhaitée.

Contrairement au perceptron, il n'est toutefois pas possible d'écrire une règle de mise à jour permettant de déterminer comment modifier tous les paramètres du réseau. Un algorithme itératif, appelé *rétropropagation du gradient*, va déterminer couche après couche, comme dans la phase de propagation avant, comment les corriger. La seule différence est que, cette fois-ci, les calculs sont effectués en commençant par la dernière couche et en

remontant jusqu'à la première. On parle de phase *backward*. C'est en effet sur la couche de sortie que l'on peut comparer l'étiquette de référence avec la prédiction.

Pour corriger sa prédiction en ne considérant que la dernière couche, le réseau dispose de deux leviers d'action : il peut modifier la matrice de paramètres à partir de laquelle les scores sont calculés, mais aussi la représentation cachée déterminée par la couche précédente. La première étape de la rétropropagation modifie donc à la fois une partie des paramètres du réseau (la matrice de la dernière couche) et permet de déterminer la représentation cachée qui aurait dû être créée par la couche précédente. Cette représentation cachée *idéale* peut ensuite servir de référence pour mettre à jour les paramètres de l'antépénultième couche, ainsi que la représentation cachée en entrée de celle-ci.

Les erreurs sont ainsi rétro-propagées dans l'ensemble du réseau, couche par couche, en remontant dans le réseau, de la première à la dernière couche. À chaque couche, la représentation cachée déterminée est comparée à la représentation cachée de référence calculée à partir de la couche suivante (ce processus équivaut techniquement au calcul d'un gradient). Cette comparaison permet de mettre à jour la matrice de paramètres de la couche actuelle et de calculer la représentation de référence pour l'étape suivante de la mise à jour.

Le principe de la rétropropagation peut sembler relativement simple, mais sa mise en pratique est très délicate et soulève de nombreux problèmes techniques. Au cours des dernières années, des progrès significatifs ont été accomplis dans ce domaine, avec l'émergence de méthodes d'optimisation spécifiquement conçues pour les réseaux de neurones, en mesure de déterminer les paramètres minimisant les erreurs.

Cependant, pour que l'algorithme d'optimisation réussisse à trouver des paramètres du réseau capables de généraliser à de nouveaux exemples, un point essentiel pour considérer

l'apprentissage comme un succès, de nombreux *hyper-paramètres* doivent être réglés manuellement et ne sont donc pas déterminés automatiquement à partir des données. Ces hyper-paramètres sont définis *en dehors* de la classe de fonctions (qui, elle, définit les paramètres) et sont fixés avant l'étape d'apprentissage. Ils correspondent, entre autres, au choix de l'architecture du réseau de neurones et aux réglages de l'algorithme d'optimisation.

Contrairement aux fonctions linéaires, les réseaux de neurones sont extrêmement sensibles aux valeurs des hyper-paramètres. Il est fréquent que l'apprentissage d'un réseau de neurones échoue en raison d'un hyper-paramètre, tel que le pas de gradient, ayant une valeur trop élevée, par exemple 0,05, alors qu'il aurait dû être de 0,04. Aujourd'hui, l'apprentissage d'un modèle nécessite toujours de nombreuses tentatives pour ajuster finement les paramètres les plus sensibles.

Trouver les bons réglages de l'algorithme d'optimisation nécessite une expérience considérable et un véritable savoir-faire. C'est pourquoi l'apprentissage des réseaux de neurones a longtemps été considéré comme une tâche complexe réservée à des spécialistes. En 2014, Demis Hassabis, le directeur de DeepMind, la société derrière AlphaGo, le premier programme capable de battre un être humain au jeu de go, qualifiait même de « magiciens » les ingénieurs responsables de l'apprentissage de ce programme⁵.

Ces dernières années ont vu émerger une accumulation d'expérience parmi les spécialistes du TAL et de l'IA, ainsi que des avancées dans notre compréhension des méthodes d'optimisation. Toutes ces connaissances ont permis de documenter de manière précise les bonnes pratiques pour l'apprentissage des modèles neuronaux, et de démocratiser l'accès à ces méthodes, même si l'apprentissage des réseaux de neurones nécessite des ordinateurs particuliers.

Des GPU pour accélérer les calculs

Comme nous venons de le voir, aussi bien la phase de propagation avant (celle qui permet au réseau de faire des prédictions) que la phase de rétropropagation (qui permet de mettre à jour les paramètres du réseau) nécessitent un grand nombre de multiplications de matrices. Cette opération qui s'écrit en quelques caractères dans les équations formalisant les réseaux de neurones demande de nombreuses instructions pour être réalisée par un ordinateur et dont l'exécution est particulièrement lente.

Cependant, il existe une astuce pour accélérer les opérations sur les matrices et, par conséquent, l'apprentissage et l'inférence des réseaux de neurones. Au lieu d'effectuer ces calculs sur le microprocesseur central CPU (*central processing unit*), le cœur d'un ordinateur chargé d'exécuter les instructions, il est possible de les effectuer sur une unité de traitement graphique, un GPU (*graphical processing unit*), un microprocesseur spécialisé dans la gestion et l'accélération des calculs liés à l'affichage graphique et qui, pendant longtemps, a été utilisé pour la modélisation 3D, le montage vidéo ou les jeux vidéos.

Par un heureux hasard, il s'avère que les calculs nécessaires pour afficher et manipuler des images, notamment en 3D, peuvent être décrits comme des opérations matricielles. Les GPU disposent d'une architecture spécialisée permettant d'effectuer ces opérations bien plus rapidement qu'un CPU, qui est plus polyvalent mais moins rapide.

En détournant l'usage premier des GPU pour leur faire effectuer les calculs nécessaires à un réseau de neurones, les chercheurs ont réussi à réduire significativement leur temps d'apprentissage et d'inférence. Plus important encore, cette réduction du temps d'apprentissage a permis de considérer des corpus de plus en plus volumineux pour entraîner les modèles. De nos jours, il est impossible (et inconcevable) d'entraîner un réseau de neurones sans disposer d'un GPU (voire de plusieurs) : les temps de calculs

seraient tellement longs que les chances de voir l'apprentissage aboutir seraient littéralement nulles.

Avoir accès à des GPUs performants est donc une condition nécessaire pour concevoir et utiliser les réseaux de neurones au cœur des giga-modèles. La révolution de l'apprentissage profond a entraîné une demande croissante pour ces puces, pourtant très coûteuses. Une seule puce A100 coûte aujourd'hui près de 10 000 dollars, et le tout nouveau H100, développé spécifiquement pour répondre aux besoins de l'IA, coûte plus de 40 000 dollars. La demande en GPUs ne faiblit pourtant pas, faisant régulièrement craindre une pénurie de puces. Nvidia, leader incontesté des GPUs avec plus de 85 % des parts de marché, peut être vu comme l'un des grands vainqueurs des progrès de l'IA : le cours de son action a plus que doublé en 2023 et la capitalisation de l'entreprise dépasse maintenant les 1 000 milliards de dollars⁶.

Un peu d'histoire

Les réseaux de neurones sont une technologie particulièrement ancienne qui a fait l'objet de nombreux travaux au sein de différentes communautés. L'idée d'utiliser un ordinateur pour reproduire le fonctionnement du cerveau humain remonte à 1943, lorsque le neuroscientifique Warren McCulloch et le mathématicien Walter Pitts ont jeté les bases de la simulation informatique des réseaux de neurones. Si cette idée paraît aujourd'hui naturelle, il suffit d'imaginer à quoi ressemblaient les ordinateurs de l'époque pour comprendre à quel point elle était révolutionnaire.

Les méthodes d'apprentissage efficaces, bien que plus récentes, ont été développées depuis la fin des années 1980. L'algorithme de rétropropagation du gradient repose sur une longue série de découvertes, mais il a été formalisé pour la première fois dans la thèse de Yann Le Cun en 1987. Cependant, il a fallu attendre les

années 2010 pour que ces méthodes soient largement adoptées et révolutionnent plusieurs domaines.

Le succès des réseaux de neurones fut tel qu'en 2018, le prix Turing fut décerné à trois scientifiques : Yoshua Bengio, Geoffrey Hinton et Yann Le Cun, reconnus par la communauté scientifique comme les *pères* des réseaux de neurones pour « les percées conceptuelles et techniques qui ont fait des réseaux neuronaux profonds un élément essentiel de l'informatique »⁷. Le communiqué de l'ACM, la société savante qui décerne ce prix, justifie l'attribution de ce prix en affirmant que les méthodes développées par ces trois scientifiques ont permis de réaliser des percées spectaculaires dans quatre domaines (la vision artificielle, la reconnaissance de la parole, le TAL et la robotique), et ont également fourni de nouveaux outils aux scientifiques de nombreux domaines allant de la médecine à l'astronomie, en passant par les sciences des matériaux⁸.

Plusieurs explications ont été avancées pour comprendre pourquoi il a fallu autant de temps pour que ces méthodes soient utilisées. Parmi les raisons souvent citées figurent en bonne place la puissance de calcul insuffisante, les corpus trop petits pour permettre l'apprentissage des représentations cachées, la complexité des algorithmes d'apprentissage, mais également les réticences des chercheurs, plus enclins à utiliser des méthodes comme les perceptrons ou les SVM, issues de la théorie de l'apprentissage statistique, qui disposaient de garanties théoriques solides et bien plus faciles à mettre en œuvre. Il faut bien reconnaître que pendant de nombreuses années, les performances des réseaux de neurones étaient nettement inférieures à celles des autres méthodes d'apprentissage.

Toujours est-il que, les réseaux de neurones ont longtemps été le vilain petit canard de l'apprentissage statistique. Leur retour sur le devant de la scène a été particulièrement étonnant et il est, aujourd'hui encore, difficile d'expliquer pourquoi, sans qu'il n'y

ait eu d'avancées conceptuelles majeures ni de découvertes significatives, ils ont soudain fonctionné et permis d'obtenir les résultats exceptionnels que l'on connaît aujourd'hui.

Un réseau de neurones pour représenter les mots : word2vec

Les réseaux de neurones sont des classifieurs *comme les autres*, qui peuvent donc être utilisés à la place des classifieurs linéaires dans les différentes applications que nous avons vues dans le chapitre précédent : identification de spam, analyse de sentiments, stylométrie...

Leur véritable force réside toutefois dans leur capacité à déterminer automatiquement les représentations les plus adaptées des entrées pour résoudre une tâche donnée. Leur intérêt est donc très limité lorsqu'ils sont appliqués à des problèmes ou à des données pour lesquels des experts disposent des connaissances suffisantes pour identifier précisément les informations à considérer pour prendre une bonne décision, comme dans l'exemple de la stylométrie (voir p. 66).

En revanche, ils révèlent tout leur intérêt et leur puissance lorsqu'ils sont appliqués sur des données brutes pour lesquelles il n'existe pas (ou peu) de connaissances préalables. C'est particulièrement vrai pour les données textuelles (celles qui nous intéressent ici), mais aussi pour les enregistrements audio ou les images. Les réseaux de neurones sont capables de faire émerger automatiquement des représentations de leurs entrées souvent bien meilleures que celles patiemment conçues par des ingénieurs, comme le relevait Yann Le Cun avec une pointe d'amertume dans une interview au journal *Le Monde*⁹ : « La technologie du *deep learning* apprend à représenter le monde. Avant, il fallait le faire à la main, expliquer à l'outil comment transformer une image afin de la classifier. Avec le *deep learning*, la machine apprend à le faire

elle-même. Et elle le fait beaucoup mieux que les ingénieurs, c’est presque humiliant ! »

La révolution ImageNet

Le premier succès des réseaux de neurones a eu lieu en 2012, dans le domaine de la vision par ordinateur, lorsque cette méthode a remporté, de manière magistrale, la compétition ImageNet. Même s’il n’est pas directement lié à l’objet de ce livre, comprendre les raisons de cette réussite est utile, car elle reflète non seulement le fonctionnement des réseaux de neurones, mais aussi celui de la recherche !

Initié en 2006, ImageNet est, avant tout, un corpus regroupant, dans les années 2010^{[10](#)}, plus d’un million d’images variées qui ont été annotées de manière collaborative et rangées dans un certain nombre de classes (plus de 2 000 dans les dernières versions du corpus) : *ballon, chien, framboise*, etc. Ces catégories permettent de faire des distinctions parfois très fines : ImageNet distingue ainsi entre 100 et 120 races de chiens. La taille d’ImageNet dépasse de loin celles des corpus utilisés jusqu’alors par les chercheurs : par exemple, le corpus Pascal VOC^{[11](#)}, développé dans les années 2000, ne comporte que 60 000 images et une vingtaine de classes.

La collecte d’un corpus aussi vaste est devenue réalisable grâce à l’abondance des données en ligne et au développement de services comme Amazon Mechanical Turk, dont nous avons déjà souligné le rôle essentiel dans les progrès de l’IA. Les images ont été collectées en effectuant des recherches ciblées, à l’aide de moteurs de recherche d’images comme Google Images, en utilisant les catégories prédéfinies. Des annotateurs ont ensuite filtré manuellement les images retournées pour valider et, en cas d’erreur, identifier la catégorie de l’objet contenu dans l’image.

L’utilisation d’images issues de l’Internet soulève divers problèmes, notamment de propriétés intellectuelles^{[12](#)} (les auteurs d’ImageNet n’ont, par exemple, pas le droit de diffuser les images

recueillies), mais a permis la constitution d'un corpus d'une taille pendant longtemps inimaginable. Plus qu'une simple opportunité technique, la création d'ImageNet repose essentiellement sur une hypothèse forte de Fei-Fei Li, l'initiatrice du projet, qui était convaincue (à raison) que pour améliorer les performances d'un classifieur, il serait plus utile de collecter et d'annoter davantage de données que de chercher à améliorer les algorithmes. Ce principe est toujours d'actualité (voir [ici](#)) : un grand nombre des succès récents de l'IA repose avant tout sur la collecte d'un gigantesque ensemble d'informations pertinentes pour la tâche envisagée.

De 2010 à 2017, le corpus ImageNet a été utilisé pour organiser l'une des principales campagnes d'évaluation en vision artificielle, dont le but était de détecter et classifier précisément des objets et des scènes dans les images naturelles. Les campagnes d'évaluation sont une pratique courante en IA, permettant d'évaluer et de comparer, dans un cadre précisément défini, les différentes approches proposées par les chercheurs pour déterminer les plus efficaces.

Lors de la campagne de 2012, l'équipe de Geoffrey Hinton, de l'Université de Toronto, proposa un système reposant, pour la première fois, sur un réseau de neurones appliqué sur les images brutes qui avait appris à la fois la représentation des images et la fonction de classification. Un tel système est généralement qualifié de « bout en bout » (*end-to-end*) pour insister sur le fait qu'il détermine automatiquement et sans intervention humaine les différentes informations dont il a besoin, et notamment la représentation de ses entrées.

Ce système a réussi, lors de la campagne de 2012, à réaliser près de deux fois moins d'erreurs que ses concurrents utilisant des méthodes d'apprentissage *classiques* avec des représentations définies manuellement (pour être précis : 16 % contre 25 % d'erreurs). Ce résultat est d'autant plus impressionnant que, dans

la plupart des campagnes d'évaluation, le score du vainqueur dépasse rarement celui du vice-champion de plus de quelques points, les systèmes comparés reposant tous plus ou moins sur les mêmes idées et proposant rarement de véritables *ruptures* technologiques.

Ce premier succès fut la preuve éclatante qu'il était possible d'apprendre des représentations d'image sans aucune intervention humaine et que le résultat dépassait largement celles conçues à la main. Cette prise de conscience validait, pour la première fois, l'intérêt des réseaux de neurones.

La méthode proposée ne reposait pas sur une nouvelle idée révolutionnaire : aussi bien l'architecture considérée (des réseaux de convolution) que la méthode d'apprentissage étaient connues depuis de nombreuses années. La mise en œuvre de ces idées fut néanmoins un véritable tour de force technologique (personne n'avait jusque-là réussi à faire fonctionner ensemble toutes les briques nécessaires) dont le succès repose aussi bien sur l'ingéniosité et le talent de l'équipe de Geoffrey Hinton que sur la disponibilité d'un corpus suffisamment grand et l'utilisation (à l'époque complètement révolutionnaire) de GPU pour accélérer les calculs, une étape nécessaire pour entraîner un modèle sur une base de cette taille. Cette victoire marqua le début du renouveau de l'intérêt pour les réseaux de neurones, non seulement parmi les chercheurs en vision, mais dans tous les domaines de l'IA. La révolution était en marche.

L'architecture de word2vec

En traitement des langues, c'est en 2013 qu'un premier modèle neuronal word2vec¹³ capable de construire automatiquement les représentations des mots marqua profondément les esprits et orienta toute la communauté ou presque vers les approches neuronales. Ce modèle fut développé par une équipe de chercheurs de Google dirigée par Tomáš Mikolov à partir des

idées qu'il avait développées lors de sa thèse à l'université de Brno, en République tchèque.

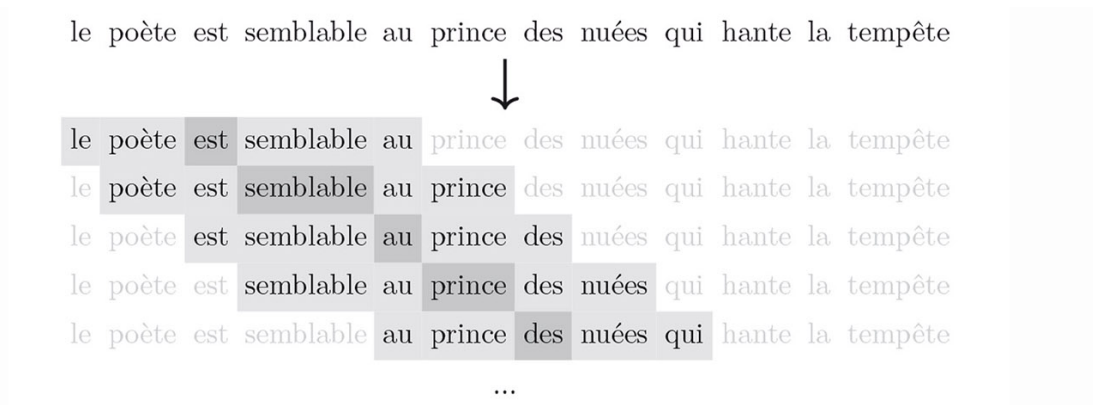
D'un point de vue purement technique, word2vec implémente un *modèle de langue* grâce à un réseau à propagation avant avec une couche cachée. Un modèle de langue est un programme capable d'estimer la probabilité d'observer un mot donné à la suite d'un début de phrase. Il est, par exemple, capable de prédire qu'il y a, à la suite de « au milieu du miroitement des vagues », une probabilité de 23 % de trouver le mot « noires » et de seulement 1 % de trouver le mot « réfléchi » (qui ne respecte pas les règles d'accord). Un modèle de langue peut, entre autres, être utilisé pour prédire le mot qui va suivre un préfixe : il suffit pour cela de calculer la probabilité de tous les mots connus (une opération potentiellement très longue) pour trouver celui avec la probabilité la plus élevée (comme pour les scores d'un modèle linéaire).

Cette tâche, d'apparence artificielle, a joué un rôle essentiel dans le développement des premiers systèmes de reconnaissance de la parole et de traduction automatique : c'est en utilisant ce type d'information qu'un système de reconnaissance de la parole est, par exemple, capable de déterminer qu'il doit reconnaître *les élèves* et non *les élève*, même si ces deux fragments de phrases se prononcent exactement de la même manière. Traditionnellement, les modèles de langue reposaient uniquement sur la fréquence, estimée sur de (très) grands corpus de textes, des mots pouvant se trouver après un préfixe donné. Mais ce type d'approche souffrait des mêmes problèmes que les classifieurs linéaires, et notamment l'impossibilité de partager des informations entre mots proches (voir [ici](#)).

Un modèle de langue peut être vu comme un problème de classification multi-classe : il suffit, pour cela, de considérer un ensemble de phrases arbitraires et d'extraire toutes les suites de n mots de ces phrases. Dans la version originale de word2vec, il s'agissait de séries de 5 mots. L'objectif est ensuite de deviner l'un

de ces mots (typiquement celui au centre) à partir du contexte. Cette formulation revient donc simplement à *masquer* le mot au centre d'une suite de termes et à essayer de le retrouver.

Cette formulation a un intérêt majeur : elle permet de définir un problème de classification supervisée (l'algorithme d'apprentissage a accès, pour chaque exemple, à la bonne réponse qu'il devra être capable de prédire), mais qui ne nécessite aucune intervention humaine pour annoter les données, contrairement aux exemples que nous avons vus au chapitre 2 : les exemples nécessaires à l'apprentissage et à l'évaluation du classifieur peuvent être construits très simplement à partir de phrases *brutes*, en extrayant toutes les suites de n mots comme dans l'exemple ci-dessous (considérant, comme le cas original, des contextes de 5 mots).



Dans cet exemple, l'utilisation d'une *fenêtre glissante* va permettre d'extraire tous les segments de cinq mots consécutifs d'une phrase. Chacun de ces segments est utilisé pour construire un exemple d'apprentissage pour word2vec : le mot au centre du segment (sur fond gris foncé) constitue l'étiquette que le modèle de langue devra prédire à partir des quatre autres mots du segment (le contexte, sur fond gris clair).

Pour atteindre cet objectif, Tomáš Mikolov et ses collègues ont proposé d'utiliser comme classifieur un réseau de neurones très simple, reposant sur les différentes composantes que nous avons introduites au début de ce chapitre. Ce réseau, décrit sur la figure

ci-dessous, commence par construire une représentation cachée pour chacun des quatre mots du contexte en utilisant une première matrice de paramètres pour transformer les représentations *one-hot* (voir [ici](#)). Dans la version originale, ces représentations cachées sont de dimension 300 : chaque mot est représenté par 300 nombres déterminés automatiquement.

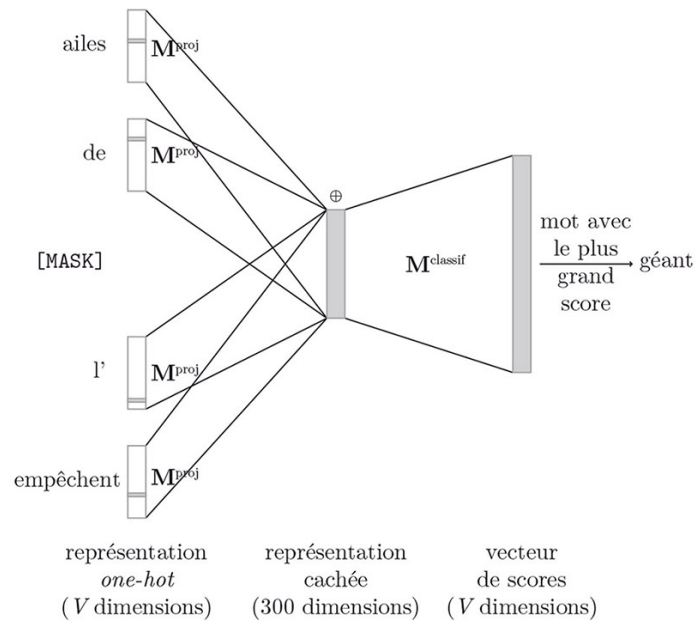


Figure 8. Le réseau de neurones implémenté par word2vec pour prédire le mot au centre d’une fenêtre. Le réseau commence par construire une représentation cachée du contexte en transformant chaque mot du contexte à l’aide d’une matrice M^{proj} en additionnant les vecteurs ainsi obtenus. Cette représentation est alors utilisée pour prédire un vecteur de score à l’aide d’une seconde matrice de paramètres, M^{score} , permettant de déterminer le mot représenté par [MASK].

Les quatre vecteurs obtenus sont ensuite additionnés pour construire la représentation cachée du contexte du mot masqué. Cette représentation est ensuite utilisée pour prédire un vecteur de scores à l’aide d’une seconde matrice de paramètres. Ce vecteur a autant de composantes qu’il y a de mots dans le vocabulaire, la i -ème valeur traduisant la confiance du modèle dans l’hypothèse « le mot masqué est le i -ème mot du vocabulaire ». Le modèle n’a alors plus qu’à chercher, comme dans le cas d’un classifieur linéaire, le score le plus grand pour choisir le mot masqué.

Il est intéressant de remarquer que ce modèle n'est pas, à proprement parler, un modèle d'apprentissage profond puisqu'il n'est composé que d'une seule couche sans aucune non-linéarité. Ce choix des auteurs de word2vec, à un moment où les chercheurs, aussi bien en vision qu'en TAL, exploraient des architectures avec de nombreuses couches cachées, est surprenant. Mais en privilégiant une architecture simple, beaucoup plus facile et rapide à entraîner, les auteurs de word2vec ont pu considérer des corpus beaucoup plus grands que ce qu'ils auraient pu faire avec une architecture plus complexe. Rétrospectivement, ce choix s'est avéré payant : la taille du corpus d'apprentissage a joué un rôle déterminant dans les bonnes performances de word2vec : comme bien souvent en TAL, la quantité de données est plus importante que la complexité (pour ne pas dire *l'intelligence*) des modèles.

L'apprentissage des deux matrices de paramètres au cœur de word2vec se fait par rétropropagation du gradient comme pour n'importe quel réseau de neurones : pour chaque exemple, le réseau prédit le mot masqué, compare sa prédiction au mot de la phrase de départ et, si ces deux mots ne sont pas les mêmes, corrige à la fois les paramètres déterminant les représentations cachées des mots du contexte et ceux calculant les scores à partir de celles-ci. C'est le principe de l'apprentissage bout à bout qui a déjà fait le succès des réseaux de neurones dans la compétition ImageNet : les représentations et le classifieur sont appris conjointement, ce qui garantit de bonnes performances, les représentations étant choisies pour faciliter la prédiction. Le modèle word2vec s'inscrit donc dans un cadre d'apprentissage supervisé : ses paramètres sont choisis de manière à reproduire les *bonnes* réponses données par le corpus d'apprentissage.

Mais contrairement au cadre introduit au chapitre précédent, l'apprentissage ne nécessite pas d'annotation spécifique à la tâche : un modèle word2vec est appris uniquement sur un corpus de phrases *brutes* dont il extrait automatiquement les informations

nécessaires pour construire des exemples étiquetés. Cet apprentissage par auto-supervision, celui des « algorithmes auto-apprenants » (*self-learning*), permet de ré-utiliser toutes les méthodes de l'apprentissage supervisé sans nécessiter l'annotation manuelle de données, et donc de considérer des corpus beaucoup plus grands. Il constitue un nouveau paradigme qui, comme nous le verrons, a joué un rôle essentiel dans le développement des giga-modèles.

À l'ère de l'Internet, ce type de ressources est particulièrement facile à trouver, même si la collecte et la gestion de grands volumes de données soulèvent plusieurs problèmes techniques. Il faut, en effet, concevoir des robots d'indexation¹⁴ capables de parcourir l'Internet et d'identifier les documents d'une langue donnée, de les télécharger en s'assurant de ne pas télécharger plusieurs fois les mêmes contenus (une tâche non triviale, le principe même de l'Internet étant de partager et de réutiliser le contenu) et d'en extraire le texte. Tout cela nécessite de disposer d'espace de stockage, de bande passante ainsi que d'une puissance de calcul non négligeables et représente un effort de développement important.

L'apprentissage de word2vec se heurte cependant à des difficultés techniques. Ces problèmes sont principalement liés à la taille du vecteur de scores en sortie du réseau, dont la dimension correspond à la taille du vocabulaire. La mise en œuvre de word2vec a nécessité l'élaboration d'une méthode d'apprentissage spécifique (appelée *échantillonnage d'exemples négatifs* – *negative sampling*) qui a permis de réduire fortement le nombre de calculs nécessaires dans la rétro-propagation du gradient et le recours à de nombreuses astuces dans sa programmation, permettant un apprentissage des paramètres en un temps raisonnable.

Grâce à tous ces efforts, les auteurs de word2vec ont pu apprendre un premier modèle sur un corpus collecté par Google (qui n'a jamais été diffusé et auquel le reste de la communauté

TAL n'a donc jamais eu accès) constitué d'un milliard de mots et avec un vocabulaire de 692 000 mots. Une seconde version du modèle¹⁵ (celle qui a été le plus utilisée), publiée de manière presque concomitante, a été entraînée sur une version étendue du corpus, comprenant 100 milliards de mots. Son vocabulaire est de 1,4 million de mots et regroupe à la fois des mots et des entités nommées comme « Le Monde » et « Paris Saint-Germain » (dans ces deux exemples, les deux termes sont considérés comme une seule entité).

Un tel corpus était sans commune mesure avec ceux répandus dans les années 2010. À titre de comparaison, les modèles de langue utilisés dans les systèmes de traduction étaient généralement entraînés sur des corpus regroupant *seulement* quelques milliards de mots, et qui paraissaient déjà gigantesques. Pour remettre la taille de ces corpus en perspective, il faut se souvenir que les 20 romans d'Émile Zola composant le cycle des Rougon-Macquart comprennent en tout et pour tout 2,6 millions de mots, l'*Encyclopædia Britannica* 44 millions et, en juillet 2023, la totalité des pages de l'encyclopédie collaborative en ligne Wikipédia à peine plus de 4,4 milliards de mots (en regroupant les pages des 280 langues)¹⁶.

L'intérêt de word2vec ne réside pas dans la capacité du réseau à résoudre la tâche pour laquelle il a été conçu (prédire un mot à partir de son contexte), mais dans les représentations cachées des mots apprises pour résoudre ce travail. Ces représentations associent chaque mot du vocabulaire à un vecteur de nombres réels. Contrairement aux représentations des mots que nous avons vues jusqu'à présent, celles-ci, appelées *plongements lexicaux* (*word embeddings*), sont de petites dimensions (quelques centaines de composantes contre plusieurs dizaines de milliers pour les représentations *one-hot*) et ne contiennent aucune composante nulle. Même si ces représentations, comme toutes les représentations cachées, sont difficilement interprétables, il est

très vite apparu qu'elles capturent des informations intéressantes liées, notamment, au sens des mots.

Une expérience, en particulier, mit en évidence ces informations et marqua profondément les esprits : la possibilité de faire des opérations mathématiques sur les plongements lexicaux. Ainsi, en soustrayant à la représentation de « Paris », le vecteur représentant « France » et en ajoutant celui représentant « Angleterre », on obtient un vecteur très proche de la représentation de « Londres ». De manière schématique :

$$\text{vect}(\text{Paris}) - \text{vect}(\text{France}) + \text{vect}(\text{Angleterre}) \approx \text{vect}(\text{Londres})$$

La notation $\text{vect}(\cdot)$ est utilisée pour souligner que les opérations ne portent pas sur les mots mais sur les représentations de ceux-ci. Malgré la simplicité de cette équation, le signe cache une opération très coûteuse en temps de calcul qui consiste à parcourir l'ensemble du vocabulaire en calculant, pour chacun des 692 000 mots que word2vec connaît, la distance entre sa représentation et le vecteur obtenu par l'opération pour trouver le mot ayant la représentation la plus proche de celle résultant de l'opération. En plus de l'aspect, il faut bien se l'avouer, un peu magique qu'il y a à pouvoir additionner et soustraire des mots entre eux, cette observation montre que des mots ayant un sens proche possèdent des représentations proches.

Cette évaluation, que l'on peut qualifier d'anecdotique (elle revient à évaluer un système de traduction par sa capacité à traduire une seule phrase, ce qui n'a guère de sens), a été confirmée par de très nombreuses expériences consistant par exemple à jauger la capacité du modèle à trouver des synonymes ou à faire des opérations comme celles que nous venons de décrire, mais cette fois-ci sur de grands corpus d'associations. Celles-ci ont montré, de manière systématique, que word2vec capturait des informations sémantiques utiles et pertinentes.

Une représentation graphique directe de ces représentations en projetant¹⁷ celles-ci dans un espace en deux dimensions (voir

[figure](#) ci-dessous) confirme ce constat. On y voit clairement émerger des « groupes » de mots ayant des sens très proches. Ainsi, tous les noms de mois sont regroupés ensemble, de même pour les noms de personnes et les noms de lieux. Ces deux derniers exemples montrent que l'espace est structuré de manière hiérarchique : le groupe des noms de personnes est bien distinct de tous les autres groupes, mais il peut également être divisé en deux sous-groupes, le premier englobant plutôt les prénoms, le second les noms propres. Il en est de même pour les noms de lieux, dont le groupe peut lui-même être divisé en trois sous-catégories : nom de pays, nom de villes et nom d'états.

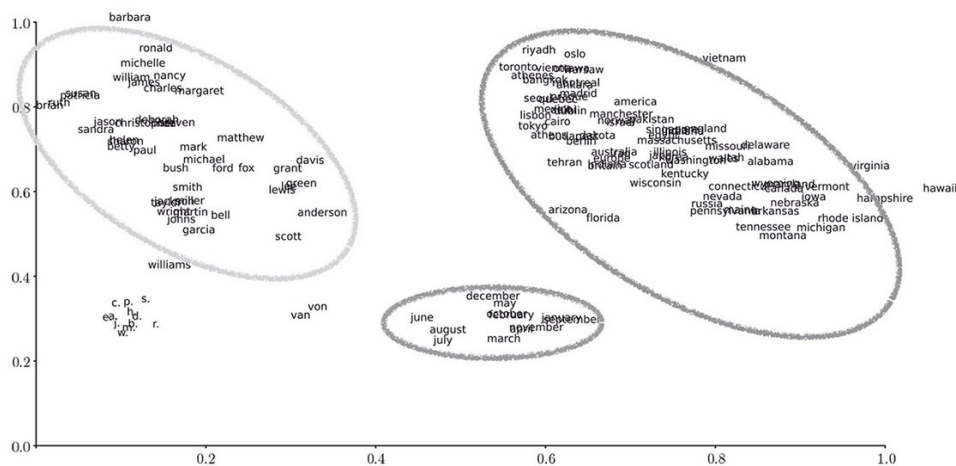


Figure 9. Projection des représentations de word2vec dans un espace à 2 dimensions. Les mots ayant un sens proche ont des représentations proches et forment des groupes dans cette visualisation.

Cette propriété des plongements lexicaux a un intérêt gigantesque : si deux mots ont des représentations proches, il est probable que changer l'un par l'autre en entrée du classifieur ne modifie pas la décision (qui, *in fine*, repose sur une combinaison linéaire des représentations). Les informations apprises par le réseau de neurones pour le mot « chaton » pourront donc être utilisées directement si le réseau *voit* le mot « chats » ou « félin » : le résultat du produit entre les représentations de ces mots et la matrice de score sera très proche. Il est donc possible, avec ces plongements lexicaux, de partager de l'information entre

caractéristiques et ainsi de réduire le nombre d'exemples nécessaires à l'apprentissage.

La capacité de découvrir des représentations pertinentes des mots sans qu'aucune information ne soit fournie au réseau (rappelons-le, le modèle est appris uniquement à partir de phrases brutes) peut paraître surprenante. Mais elle s'explique très facilement grâce à une théorie linguistique, l'hypothèse distributionnelle, formulée dans les années 1950, notamment par Martin Joos, Zellig Harris et John Firth. Selon ces linguistes, deux mots qui apparaissent dans un même contexte tendent à avoir des sens proches. L'intuition derrière cette hypothèse apparaît clairement en essayant de trouver le mot masqué dans la phrase suivante :

« Bingzhi aime les pizzas et Nour le [MASK]. »

S'il existe une infinité de termes pouvant prendre la place du mot masqué, toutes les possibilités partagent deux traits : ce sont des noms masculins décrivant plutôt des aliments. La représentation du mot masqué construite à partir du contexte devra donc capturer ces caractéristiques communes et les traduire en nombres.

La qualité des plongements lexicaux de word2vec n'est pas la seule raison qui explique le succès de ce modèle. Les auteurs de la méthode ont été les premiers à diffuser les plongements lexicaux qu'ils avaient découverts sur un très gros corpus, et pas seulement le code source de leur modèle (qui permet à n'importe qui d'apprendre un modèle sur ses propres données).

Tous les chercheurs en TAL ont dès lors pu les utiliser très facilement dans leurs expériences sans avoir besoin de télécharger des corpus de très grande taille ni de réentraîner un modèle, deux opérations chronophages. Très vite, les plongements lexicaux sont devenus une brique essentielle du TAL. De nombreuses expériences ont montré que, en règle générale, leur utilisation améliore significativement les performances et simplifie le développement des classifieurs.

Les plongements lexicaux ont également ouvert la porte à de nouvelles manières d'analyser les textes : ils ont par exemple été utilisés par des chercheurs en sciences humaines pour modéliser l'évolution du terme « islamo-gauchiste »¹⁸ ou définir un mouvement littéraire comme le romantisme¹⁹. Preuve de cette popularité (selon les critères généralement utilisés pour évaluer l'impact d'un travail scientifique), les articles introduisant word2vec avaient déjà été cités plus de 3 000 fois en 2015 (l'année qui a suivi leur publication) et près de 81 000 fois au moment où ces lignes sont écrites. À titre de comparaison, une étude²⁰ portant sur l'ensemble des travaux en TAL publiés entre 1965 et 2015 montre que l'auteur le plus cité, Herman Ney, a publié dans ce laps de temps 343 articles qui ont recueilli, au total, 5 200 citations. La diffusion de ce qui fut bientôt appelé un *modèle pré-entraîné*, une pratique désormais courante dans le TAL et au cœur des giga-modèles, explique, en très grande partie, la popularité de word2vec. Si l'idée de l'auto-supervision avait été introduite dès 2003 par l'équipe de Yoshua Bengio à l'université de Montréal, et si deux chercheurs qui travaillaient alors au laboratoire de NEC, Ronan Collobert et Jason Weston, avaient montré dès 2008 qu'il était possible d'utiliser les plongements lexicaux appris par un modèle de langue dans de nombreuses tâches de TAL, aucun de ces travaux n'a eu l'impact de word2vec.

La révolution des transformeurs

En dépit de leurs succès, les plongements lexicaux de word2vec pèchent par leur caractère *statique* : ils sont précalculés une fois pour toutes à partir des données d'apprentissage et ne sont plus jamais modifiés par la suite. Une conséquence immédiate est que la représentation d'un mot déterminée par word2vec ne dépend pas du contexte dans lequel celui-ci est utilisé, de sorte que deux homonymes auront toujours exactement la même représentation. Par exemple, dans les deux phrases suivantes :

J'ai mangé un avocat.

J'ai raconté des salades à mon avocat.

Les vecteurs représentant le mot « avocat » coïncident et ne distinguent pas la personne du fruit. Un problème similaire se pose pour les deux occurrences du mot « son » dans les phrases suivantes :

La journaliste a terminé son travail.

Le journaliste a terminé son travail.

Les représentations du mot « son » ne tiennent pas compte du fait que le possesseur (le sujet de la phrase) est féminin dans un cas et masculin dans l'autre. Cette information est pourtant vitale, par exemple, pour traduire correctement ce mot en anglais (par « *her* » dans le premier cas et « *his* » dans le second).

De nombreux réseaux de neurones ont été introduits pour apporter une réponse à ce problème et construire des représentations des mots qui dépendent du contexte : réseaux convolutifs, modèles hiérarchiques basés sur une analyse syntaxique de la phrase, réseaux récurrents, et bien d'autres encore. Le nombre et la diversité des solutions imaginées témoignent du processus de recherche itératif, par essai-erreur, qui a permis d'apporter une solution à ce problème, avec de nombreuses propositions restées sans suite et de bonnes idées parfois impossibles à mettre en œuvre. Finalement, la quasi-totalité des méthodes explorées sont aujourd'hui complètement oubliées, seuls deux modèles ayant émergé de tous ces efforts.

Le premier, les réseaux récurrents, repose sur une idée simple : la phrase est lue de gauche à droite, mot après mot. Chaque mot est associé à une représentation cachée qui est construite de manière incrémentale en combinant ce mot avec la représentation du mot précédent. Ainsi, les informations contextuelles s'accumulent dans la représentation cachée au fur et à mesure que les mots sont lus, et celle-ci peut contenir des informations provenant de tous les

mots précédents. Les réseaux récurrents ont notamment servi en traduction automatique : le vecteur représentant la phrase à traduire est passé à un second réseau qui va générer, en n'utilisant que les informations contenues dans ce vecteur, une traduction. L'idée de pouvoir ainsi *synthétiser* tout le contenu d'une phrase en quelques dizaines de nombres est assez incroyable !

Si l'idée de combiner incrémentalement la représentation des mots est intuitivement attrayante, elle présente plusieurs inconvénients. À mesure que de nouveaux mots sont lus, l'influence des mots précédents diminue rapidement, et les informations du passé sont progressivement *oubliées*. Ce comportement complique considérablement l'apprentissage des paramètres du réseau en rendant les gradients soit trop petits soit trop grands (problème dit du *vanishing/exploding gradient*).

Plus grave encore, les calculs ne peuvent être effectués que mot par mot : pour calculer la représentation du *i*-ème mot d'une phrase, il est nécessaire d'avoir calculé les représentations de tous les mots précédents. Cela limite l'exploitation des capacités de calcul des GPUs et rend l'apprentissage trop lent pour être efficace à grande échelle : les temps de calcul nécessaires pour appliquer ce modèle à de grands corpus sont tout simplement prohibitifs.

En 2017, des chercheurs de Google Research et de Google Brain, deux entités d'Alphabet se consacrant à la recherche en IA, ont introduit une nouvelle architecture, les *transformers* (en anglais), qui a marqué une étape clé dans la capacité des ordinateurs à comprendre et manipuler le langage. Comme nous le verrons dans la suite de ce livre, cette architecture joue un rôle tellement fondamental aujourd'hui en IA que son nom est quasiment devenu un nom commun. C'est pourquoi nous avons choisi d'utiliser l'orthographe française de ce nom de modèle, même si son utilisation n'est pas encore systématique.

Cette nouvelle architecture représente, une nouvelle fois, une réponse purement pratique aux problèmes techniques des réseaux

récurrents, notamment la difficulté de l'apprentissage et sa durée. En 2014, l'équipe de Yoshua Bengio (celle-là même qui avait introduit la notion d'auto-supervision), avait déjà proposé²¹ un mécanisme appelé *attention*, qui permettait à un réseau récurrent d'apprendre à se concentrer sur les parties importantes de la phrase. Bien qu'ils l'aient introduit dans un contexte très spécifique, la traduction automatique, Samy Bengio et son équipe ont rapidement montré que ce mécanisme pouvait contribuer à surmonter certaines limites fondamentales des réseaux récurrents.

Quelques années plus tard, les chercheurs d'Alphabet ont découvert que le mécanisme d'attention était si puissant qu'il pouvait complètement remplacer la construction incrémentale de la représentation cachée dans le réseau récurrent, réduisant ainsi de manière significative le temps de calcul requis pour l'apprentissage et l'inférence. Ils ont donc proposé une architecture qui se basait uniquement sur le mécanisme d'attention, en abandonnant complètement la construction incrémentale des représentations. Le titre de l'article décrivant cette architecture, *Attention is all you need*²², est on ne peut plus explicite. L'utilisation de l'attention comme unique mécanisme pour prendre en compte les informations du contexte a rendu possible le fait d'implémenter efficacement les différentes opérations nécessaires à l'apprentissage d'un modèle, en exploitant pleinement les capacités des GPUs. Une fois de plus, la simplification du modèle pour accélérer les calculs et traiter davantage de données a été la clé du succès !

Les transformeurs jouent aujourd'hui un rôle si crucial dans les modèles de TAL, et plus généralement dans toute l'IA, que nous consacrerons un chapitre entier à explorer leurs capacités (voir [ici](#)). Mais avant cela, commençons par bien comprendre leur fonctionnement.

À très haut niveau, un transformeur est essentiellement un réseau de neurones qui prend en entrée une phrase, ou plus précisément

une représentation initiale des mots d'une phrase (généralement sous forme de vecteurs *one-hot*), et qui, comme illustré sur la figure ci-dessous, génère une représentation contextualisée pour chaque mot.

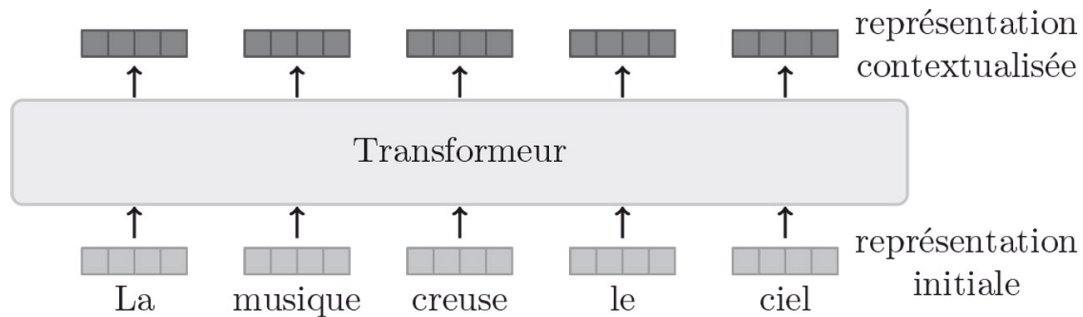


Figure 10. Représentation à haut niveau d'un transformeur : le réseau de neurones permet d'associer chaque mot d'une phrase à un plongement lexical contextualisé capturant les propriétés syntaxiques et sémantiques du mot dans le contexte dans lequel celui-ci est utilisé. D'un point de vue purement fonctionnel, le modèle transforme une séquence de vecteurs *one-hot* représentant une phrase en une séquence de même taille composée de vecteurs de petite dimension.

Le transformeur peut donc être vu comme un réseau de neurones transformant (pour une fois que le nom d'un modèle correspond à ce qu'il fait !) un ensemble de n vecteurs (dans notre exemple, chaque vecteur représente un mot) en un autre ensemble de n vecteurs. Même si elle a été initialement conçue pour les données textuelles, le principe de cette architecture est très général et peut être appliqué à n'importe quel type de données constitué d'un ensemble d'éléments : images (un ensemble de pixels), audio (un ensemble de nombres représentant l'amplitude du signal en fonction du temps), molécules chimiques (un ensemble d'atomes)²³, etc.

Un transformeur est composé de plusieurs couches, appelées *couches d'encodage*, suivant ainsi la principale intuition de l'apprentissage profond : la succession de couches va permettre de construire des représentations de plus en plus abstraites. Plus précisément, dans un transformeur, chaque couche prend en entrée n vecteurs, chacun représentant (dans le cas du TAL) un

mot de la phrase, et les enrichit en y incorporant des informations contextuelles, provenant des autres mots de la phrase. La sortie de chaque couche d'encodage comporte ainsi n vecteurs, qui deviendront l'entrée de la couche suivante, comme illustré sur la figure ci-contre.

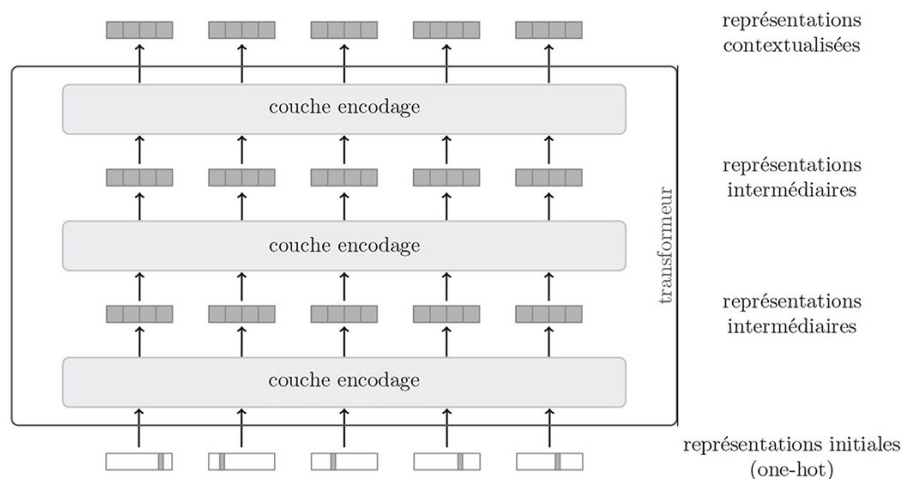


Figure 11. Aperçu de l'architecture d'un transformeur : le réseau prend en entrée une séquence de vecteurs *one-hot* qui constitue une phrase. Il applique ensuite plusieurs couches d'encodage de manière successive, construisant ainsi une série de représentations cachées intermédiaires pour chaque vecteur en entrée. La dernière séquence de représentations cachées construite correspond à la sortie du réseau.

Cette architecture définit un *encodeur*, c'est-à-dire un réseau de neurones capable de représenter une observation par un vecteur. Il est composé de plusieurs couches d'encodages. Chacune de ces couches est, en simplifiant un peu, composée elle-même de deux parties : une couche d'auto-attention et une couche à propagation avant, similaire à celle vue au début de ce chapitre (voir [ici](#)). La figure ci-contre détaille l'organisation d'une couche d'encodage.

C'est la couche d'auto-attention qui va encoder des informations provenant du contexte dans la représentation des mots en construisant celles-ci comme des combinaisons linéaires (qui méritent vraiment leur titre d'opération reine de l'apprentissage !) des représentations des autres mots de la phrase. Plus précisément, la représentation intermédiaire du i -ème mot de la

phrase, appelé *cible* dans le calcul de l'attention, est construite en additionnant les représentations de tous les autres mots (appelés *sources*) de la phrase (y compris celle du mot lui-même) après avoir transformé celles-ci à l'aide d'une matrice M^{values} . Le poids d'un mot A dans la combinaison linéaire permettant de calculer la représentation du mot B , appelé *attention*, attention de A à B , indique l'importance de A pour déterminer la représentation de B et donc, indirectement, son sens ou ses propriétés grammaticales.

L'attention entre deux mots est déterminée à partir de leurs représentations : une matrice de paramètres M^{query} transforme la représentation du mot source, et une autre matrice M^{key} transforme celle du mot cible. L'attention est alors définie comme le produit scalaire entre ces deux vecteurs (encore une combinaison linéaire !).

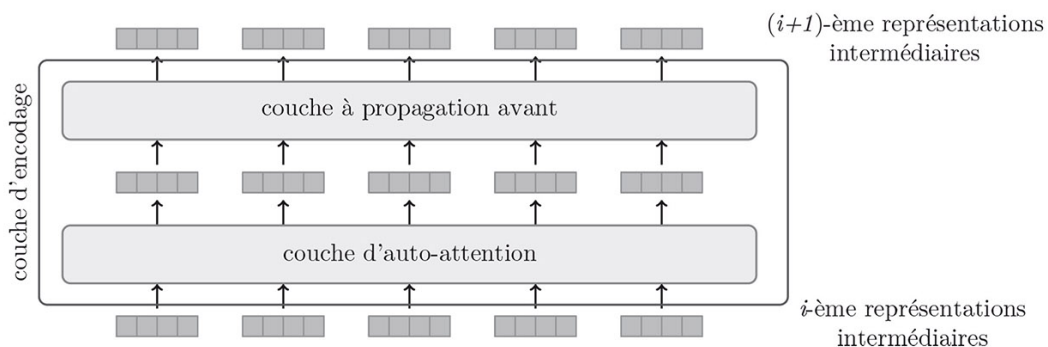


Figure 12. Les différentes composantes d'une couche d'encodage dans un transformeur. La représentation cachée construite à l'étape précédente est enrichie à l'aide de deux transformations. La première consiste à *injecter* des informations contextuelles en utilisant une couche d'auto-attention, tandis que la seconde repose sur une couche de propagation avant pour *adapter* la représentation construite à la tâche considérée. Ces deux transformations sont appliquées simultanément à tous les mots de la phrase.

Au final, chaque couche d'auto-attention va déterminer trois représentations distinctes de chacun des n mots de la phrase et calculer $n \times n$ attentions pour encoder les informations provenant du contexte. Ces trois représentations correspondent aux différents rôles qu'un mot peut jouer dans le calcul de l'attention : en tant que source, en tant que cible, et en tant que composante de

la combinaison linéaire. Cette méthode, bien que complexe et peu intuitive, résulte de multiples expérimentations. Sa justification principale, voire unique, est historique.

Comme pour tous les réseaux de neurones, tous ces paramètres sont déterminés à partir des données du corpus d'apprentissage et le transformeur va apprendre à calculer automatiquement l'attention entre deux mots, et donc à sélectionner les éléments de la phrase pertinents pour déterminer la représentation d'un mot donné.

La figure ci-dessous donne un exemple de l'attention entre le pronom anglais « *it* » et les autres mots de la phrase. Elle montre que la représentation de ce terme est fortement influencée par la représentation de ses antécédents et que, d'une certaine manière, le modèle (re)découvre les relations syntaxiques et s'appuie sur elles pour construire la représentation des mots. Ce résultat peut paraître intuitif, mais il ne faut pas perdre de vue que cette attention est déterminée automatiquement par le transformeur, sans aucune intervention humaine.

Le principe de la construction des représentations par une couche d'encodage est donc très simple, même si nous sommes toutefois concentrés ici sur les deux éléments centraux de l'architecture pour contextualiser les représentations, tout en laissant de côté plusieurs autres composantes tout aussi nécessaires, telles que les couches de normalisation, les connexions résiduelles et la représentation des positions. Mais sa mise en œuvre pratique est légèrement plus compliquée.

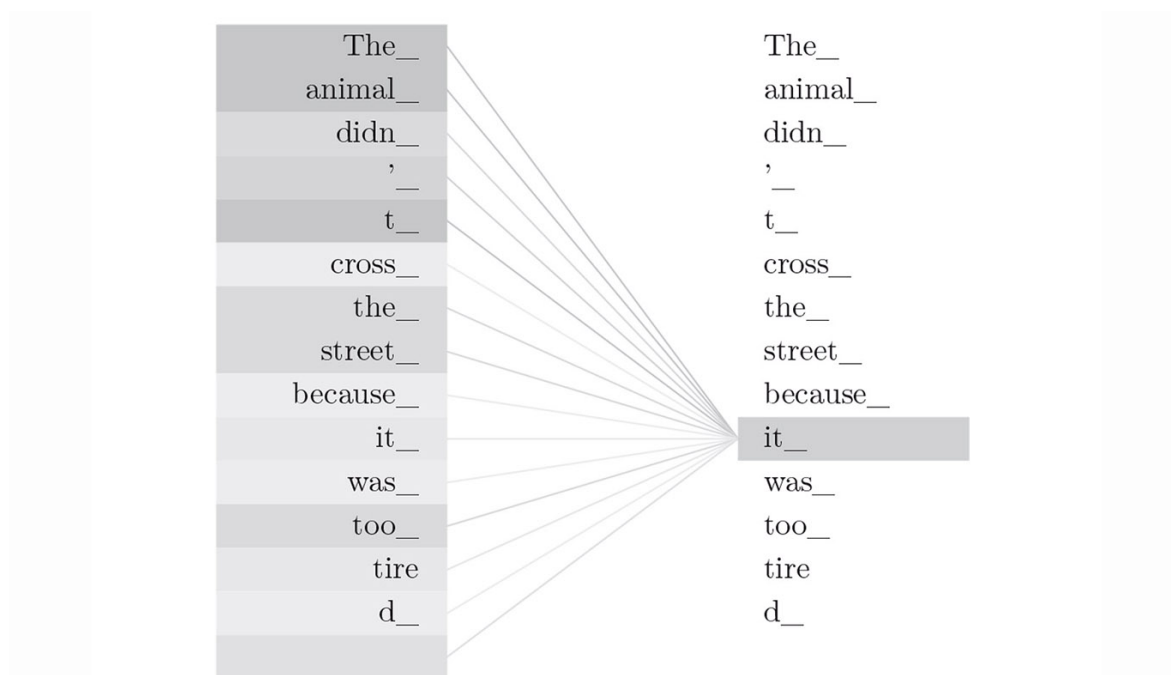


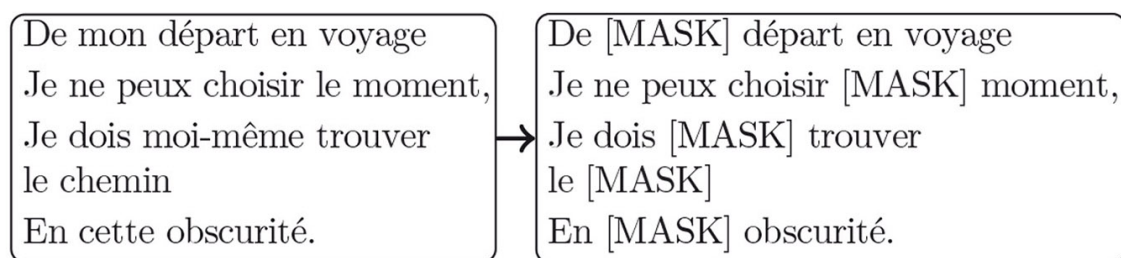
Figure 13. Visualisation de l'auto-attention pour déterminer la représentation du mot « *it* » dans la phrase « *The animal didn't cross the street because it was too tired* ». Le fond de chaque mot est d'autant plus foncé que l'attention entre ce mot et « *it* » est grande. La phrase est segmentée en sous-mots (voir [encadré](#)).

Comme nous l'avons déjà vu, un transformeur est constitué d'un empilement de couches d'encodage qui introduiront progressivement davantage d'informations contextuelles et abstraites dans les représentations. Pour réussir à construire de *bonnes* représentations, il est également nécessaire de multiplier les couches d'attention : en pratique, chaque couche d'encodage comporte plusieurs couches d'auto-attention appelées *têtes d'auto-attention* (*attention head*). Chaque tête a des paramètres distincts et calculera, indépendamment, une représentation intermédiaire qui sera ensuite transformée par la couche à propagation avant. Ainsi, une couche d'encodage construit autant de représentations qu'il y a de têtes d'attention, qui sont ensuite concaténées pour déterminer la représentation finale des mots construite par le réseau.

Le choix de multiplier le nombre de têtes d'attention, comme la plupart des décisions prises lors de la conception de l'architecture des transformeurs telle que nous la connaissons aujourd'hui,

résulte d'innombrables essais, dont beaucoup ne furent pas concluants. Bien que ces décisions aient finalement conduit à des performances exceptionnelles, il est essentiel de garder à l'esprit qu'elles ne reposent sur aucun fondement théorique ni aucune modélisation du fonctionnement du cerveau, du raisonnement ou de la manière dont les humains utilisent le langage, une approche à l'opposé de celle qui avait cours aux débuts de l'IA. Leur seule validation étant expérimentale, il n'y a pas de garantie que ces choix soient optimaux, et il est possible qu'un modèle plus simple ou offrant de meilleures performances puisse être développé.

Pour apprendre tous les paramètres d'un transformeur (les trois matrices nécessaires aux calculs des attentions de chaque tête – ceux du réseau à propagation avant, et tous ceux dont nous n'avons pas parlé), il est possible d'utiliser une méthode d'auto-supervision en tout point similaire à celle de word2vec : les paramètres sont choisis de manière à retrouver les mots masqués dans une phrase. Cette fois-ci, les exemples nécessaires ne seront pas générés en utilisant une fenêtre glissante, mais en prenant chaque phrase du corpus d'apprentissage et en masquant un pourcentage des mots de cette phrase (généralement 20 %) choisis au hasard, comme dans l'exemple suivant :



Un transformeur est donc entraîné uniquement à partir de données brutes, sans qu'aucune indication liée au sens des mots ou à leur rôle syntaxique ne lui soit donnée. Cette caractéristique est essentielle : elle permet d'apprendre à partir de corpus de très grande taille sans qu'il soit nécessaire d'annoter ceux-ci, une opération, comme nous l'avons vu, longue et coûteuse. Pour mettre en œuvre cette approche, il est nécessaire de compléter

l'architecture que nous venons de décrire pour être capable de *re-transformer* les vecteurs en sortie du réseau en mots. Il est possible, pour cela, d'ajouter une tête de classification (à ne pas confondre avec les têtes d'attention) à la sortie de l'encodeur.

Cette tête de classification est généralement un simple réseau à propagation avant de dimension $h \times V$, où h est la taille des représentations de l'encodeur et V celle du vocabulaire. Ce réseau est un modèle linéaire qui calcule la probabilité que le mot masqué considéré soit le i -ème mot du vocabulaire. Une fois la tête de classification ajoutée, il est possible d'entraîner un classifieur en utilisant, comme pour tous les réseaux de neurones, une descente de gradient stochastique et une rétropropagation du gradient pour déterminer les mises à jour à apporter aux paramètres : le modèle prédira le mot le plus probable pour chaque position masquée, puis en comparant le mot prédit avec le mot original, il identifiera les erreurs et mettra à jour les paramètres pour augmenter les chances que le mot original soit prédit si l'exemple est présenté une nouvelle fois au réseau.

On a beaucoup dit que ces modèles apprenaient à comprendre des phrases (la compréhension étant ici limitée à la construction de représentations des mots) uniquement en lisant des textes. Il me semble plus juste d'affirmer que les transformeurs prédisent des symboles (des suites de caractères arbitraires) en connaissant d'autres symboles : leur seule compétence est de modéliser les liens entre les symboles du corpus d'apprentissage et de les exploiter pour produire et, plus précisément, compléter des suites de symboles.

En particulier, il n'est pas complètement correct de dire que ce modèle découvre ou apprend le sens du mot « verte ». En tout cas, il est incapable d'établir le rapport entre le nom de la couleur et sa réalité physique (sa position dans le spectre visible des couleurs) : si l'on remplaçait toutes les occurrences de « verte » par « anaérobique » (ou toute autre séquence de caractères) dans le

corpus d'apprentissage, le modèle produirait sans hésitation des phrases comme « l'aurore grelottante en robe rose et anaérobique », puisqu'il ne ferait aucune distinction entre les deux termes et les emploieraient indifféremment l'un ou l'autre, au grand dam de la physique.

Le principe des transformeurs a été initialement introduit dans le domaine de la traduction automatique. Le modèle que nous venons de décrire était alors couplé à un second réseau, le *décodeur*, ayant pour rôle de générer la phrase dans la langue cible. Nous reviendrons très vite sur ce deuxième type de réseau (voir [ici](#)). Plus que le modèle lui-même ou la qualité des traductions générées, c'est la mise à disposition d'un modèle pré-entraîné l'année suivante qui a déclenché un véritable raz-de-marée dans la communauté TAL : en 2018, une seconde équipe de Google a diffusé ce qui sera bientôt appelé un *modèle de langue* permettant à quiconque d'obtenir des représentations vectorielles des mots d'une phrase en anglais, sans nécessiter un corpus de grande taille, des ressources de calculs importantes ou une expertise particulière.

Ce modèle appelé *BERT* (*Bidirectional Encoder Representations from Transformers*) est constitué de 24 couches d'encodage, chacune comportant 16 têtes d'attention, totalisant près de 340 millions de paramètres, qui ont été estimés sur un corpus contenant des articles de Wikipédia (2,5 milliards de mots) et d'un corpus de livres (800 millions de mots). Jamais auparavant un modèle avec autant de paramètres n'avait été développé ! Rapidement, des modèles similaires ont été créés pour les principales langues présentes sur l'Internet : dès 2019, des modèles pré-entraînés utilisant exactement la même architecture que BERT ont été rendus disponibles pour des langues telles que le finnois (FinBERT), le néerlandais (BERTje), le basque (BERTeus), afrikaans (AfriBERT) et bien d'autres. Pour le français, deux modèles ont été développés : FlauBERT par un consortium de trois universités, et CamemBERT par l'INRIA.

En moins d'un an, des modèles de type BERT ont été intégrés à presque tous les systèmes de TAL, offrant systématiquement des améliorations de performance significatives, quelle que soit la langue ou la tâche considérée, et rendant immédiatement caducs de nombreux travaux de recherche qui s'efforçaient d'élaborer des représentations adaptées à des tâches spécifiques. BERT a été utilisé, par exemple, dans des tâches de classification telles que l'analyse de sentiments, dans les systèmes de question-réponse et de résumé automatique, mais également pour compléter automatiquement des textes.

En octobre 2019, Alphabet a commencé à utiliser BERT pour analyser les requêtes en anglais sur son moteur de recherche, Google. Quelques mois plus tard, en décembre, le modèle était capable d'analyser les requêtes dans plus de 70 langues. Les résultats de ces premiers tests ont été si prometteurs qu'un an plus tard, en décembre 2020, l'entreprise annonçait que BERT traitait l'intégralité des requêtes en anglais sur son moteur de recherche, démontrant ainsi la capacité exceptionnelle du modèle à comprendre la langue.

Segmentation en unités sous-lexicales

Les réseaux de neurones utilisent des vecteurs pour représenter les mots et effectuent des opérations avec ces vecteurs. Il y a seulement deux moments où ils doivent associer ces représentations continues à des mots : lors de l'entrée dans le réseau et en sortie.

Ces étapes posent deux problèmes. D'abord, les réseaux ne peuvent pas gérer les mots inconnus, car ils ne savent pas comment les représenter. De plus, la taille du vocabulaire (le nombre de mots connus) a un impact sur la complexité des calculs, et peut rendre le processus particulièrement long. Afin de résoudre ces problèmes, des chercheurs ont développé des algorithmes pour décomposer les mots en parties plus petites appelées *unités sous-lexicales*, *sous-mots* ou *tokens*.

Ces algorithmes reposent sur la même intuition que la méthode que nous avons apprise à l'école pour construire le sens d'un mot en identifiant sa

racine, ses préfixes et ses suffixes : le sens du mot « anticonstitutionnellement » peut ainsi être déduit de sa décomposition en un radical, l'adjectif « constitutionnel » ; un préfixe (*anti-*) marquant l'opposition, et un suffixe (*-ment*) utilisé pour former des adverbes. Cette décomposition permet de construire le sens et la représentation des mots inconnus, mais aussi de réduire la taille du vocabulaire.

En pratique, cette décomposition ne repose pas sur une analyse linguistique, mais se fait de manière complètement statistique en identifiant les suites de caractères les plus fréquentes. Les méthodes utilisées aujourd'hui comme BPE ou SentencePiece, construisent le vocabulaire directement à partir de l'analyse du corpus d'apprentissage utilisé pour déterminer les paramètres d'un modèle en utilisant un algorithme itératif très simple. Au départ, chaque caractère est considéré comme un token autonome. Ensuite, l'algorithme identifie la suite de deux tokens la plus fréquente et les fusionne en un seul. Ce processus est répété jusqu'à atteindre une taille de vocabulaire prédéfini. Les tokens ainsi obtenus correspondent à des mots complets lorsque les mots sont fréquents, tandis que les mots plus rares sont décomposés en groupes de caractères.

Les réseaux de neurones et, en particulier, les modèles de langues, ne voient donc pas directement le même texte que nous, mais plutôt une suite de sous-mots correspondant plus ou moins à notre notion de mots. Voici par exemple comment un modèle BERT *voit* le début du poème *L'Albatros*, de Charles Baudelaire :

So _ uve _ nt _ , _ pour _ s _ ' _ am _ use _ r _ , _ les _
hommes_d'_équipage_prennent_des_alba_tros_,_vaste_s_oiseau
x_des_mer_s_,_qui_suivent_,_indo_lent_s_compagno_ns_de_voyag
e_,_le_navire_gli_ssant_sur_les_gouf_fre_s_am_ers_..

Si les mots fréquents (« équipage », « oiseaux », etc.) sont correctement identifiés, ce n'est pas le cas des mots plus rares et, bien souvent, le modèle devra commencer par assembler des suites de caractères dont le sens est difficile à déterminer (par exemple « fre » ou « tros ») pour retrouver les mots.

La segmentation en sous-mots est souvent considérée comme un détail technique, mais elle a un impact majeur sur le fonctionnement des modèles. Il est essentiel de comprendre que les systèmes de TAL actuels manipulent principalement des groupes de caractères plutôt que des

mots complets, même si la plupart des explications des réseaux de neurones, y compris celles que l'on trouve dans ce livre, confondent délibérément mots et sous-mots pour des raisons pédagogiques. Cela implique en effet que les réseaux de neurones doivent apprendre à composer des groupes de caractères pour reconstruire le sens des mots avant de les utiliser pour comprendre les phrases. Comme le nombre de sous-mots résultant de la segmentation d'un mot est directement lié à la fréquence de ce mot dans les données d'apprentissage, les mots rares ou moins fréquents sont, intuitivement, moins bien modélisés et apparaissent moins souvent dans les textes générés par les grands modèles. De nombreuses expériences confirment cette intuition et suggèrent que la segmentation peut contribuer à des biais dans les prédictions des modèles. Par exemple, les formes féminines de certains noms de métiers sont très rares et donc moins bien représentées.

Analyser les représentations cachées

Pourquoi les transformeurs réalisent-ils des prouesses ? Il est difficile de répondre, car la qualité des décisions qu'ils permettent de prendre se fait au prix de la difficulté de les interpréter. En effet, les représentations des transformeurs sont *cachées* : elles ont été choisies automatiquement uniquement dans le but de permettre au réseau de retrouver les mots masqués, sans qu'on ne dispose d'un contrôle direct sur les informations qui y sont encodées.

Pour expliquer les bonnes performances des transformeurs, les chercheurs ont rapidement proposé de nombreuses méthodes pour identifier les informations capturées dans les représentations de BERT et déterminer à quel point elles encodaient des informations linguistiques. Il est intéressant de noter que plusieurs de ces méthodes s'inspirent directement des méthodes de psycho-linguistique utilisées pour comprendre comment les humains représentent et utilisent le langage : les modèles ont atteint une telle complexité qu'ils sont désormais étudiés avec des méthodes similaires à celles utilisées pour étudier les processus

cognitifs mis en œuvre dans le traitement et la production du langage.

L'étude des représentations de BERT est devenue un sujet de recherche si important qu'un terme, la *bertologie* a été inventé pour le désigner. L'article²⁴ qui a introduit ce terme, publié en 2020, soit moins de deux ans après la publication de BERT, répertorie déjà 150 articles traitant de cette question ! Bien qu'il soit impossible de synthétiser tous ces travaux, on peut affirmer qu'ils montrent qu'il n'y a aujourd'hui que très peu de doute quant au fait que les transformeurs parviennent à apprendre, sans aucune intervention humaine, une quantité substantielle d'informations linguistiques, que ce soit au niveau morphologique, syntaxique ou sémantique.

Parmi les nombreuses méthodes proposées pour analyser les représentations découvertes et utilisées par BERT, l'une des plus intéressantes et représentatives est la tâche d'accord sujet-verbe. Cette tâche consiste simplement à demander à un modèle de choisir entre deux formes d'un verbe celle qui est la plus probable. Pour s'assurer que le modèle a appris des informations linguistiques, il suffit de vérifier si sa prédiction respecte bien les règles grammaticales de la langue considérée, notamment la règle d'accord en nombre. Par exemple, si un modèle est capable de prédire que, dans la phrase « Les chiens [MASK] tous les soirs », il est plus probable que le mot masqué soit « aboient » plutôt que « aboie », alors on peut en conclure qu'il a appris la règle d'accord entre le sujet et le verbe.

Ce test revêt une grande importance d'un point de vue linguistique : la capacité à accorder le verbe avec son sujet est souvent utilisée pour démontrer qu'une personne (ou, dans notre cas, un réseau de neurones) construit une représentation hiérarchique d'une phrase. En effet, bien que la prédiction de l'accord dans l'exemple précédent soit triviale (un corpus assez grand contient de nombreuses occurrences de « chiens aboient » et sûrement aucune de « chiens aboie »), il est possible de

complexifier le test en éloignant arbitrairement le sujet du verbe, comme dans l'exemple suivant :

« Les chiens en train de jouer dans le jardin du voisin [MASK]. »

Dans ce cas, l'accord ne se fait pas avec les mots dans le voisinage immédiat du verbe et il est nécessaire, pour prédire la forme correcte de celui-ci, de reconstruire la structure syntaxique de la phrase. Arriver à prédire ce type d'accord est une belle preuve que le modèle ne se contente pas de manipuler des formes de surfaces de manière quasi-automatique en exploitant simplement des corrélations entre symboles mais, au contraire, qu'il construit (et manipule) une représentation abstraite des phrases dans laquelle le sujet (« chien ») est directement relié au verbe, même si ceux-ci ne sont pas l'un à côté de l'autre dans la phrase.

De nombreuses expériences ont montré que, même dans les cas les plus complexes, les transformeurs sont capables d'accorder, avec une précision stupéfiante, le verbe avec le sujet, mais également, dans le cas du français, le participe passé avec le complément d'objet. Ce résultat est absolument remarquable ! Il faut, en effet, se souvenir que ce modèle a été appris de manière complètement auto-supervisée : il n'a jamais reçu aucune indication lui expliquant qu'il y a, dans une phrase, des verbes et des noms (entre autres), que ces derniers peuvent se conjuguer au singulier ou au pluriel, que certains noms jouent un rôle spécial en tant que *sujets* du verbe, et que le sujet et le verbe s'accordent en nombre. Il a donc dû découvrir de manière complètement autonome et sans intervention humaine toutes ces règles simplement en jouant au jeu « prédit le mot caché », sans lien apparent avec la linguistique.

Les résultats de cette expérience, confirmés par de nombreux autres, montrent que la qualité des prédictions des transformeurs peut s'expliquer par le fait qu'ils capturent, dans les représentations qu'ils manipulent, des informations linguistiques

similaires à celles qui permettent de comprendre et de modéliser la manière dont les humains construisent le sens d'une phrase. Toutefois, il serait hâtif de conclure que les transformeurs analysent les phrases en suivant les mêmes processus que les humains : Najoung Kim, une chercheuse de l'Université de New-York, et son collègue Tal Linzen ont mis en évidence, au travers d'une série d'expériences remarquables, que les transformeurs ne possèdent pas la capacité de composer les structures linguistiques de la même manière que nous.

Les humains sont capables d'exploiter la nature compositionnelle inhérente au langage pour construire le sens d'une déclaration en se basant sur le sens des éléments qui la composent. C'est cette capacité qui explique qu'un être humain puisse comprendre, bien que parfois avec un certain effort, des structures complexes, comprenant de nombreuses imbrications, telles que « le père du frère de la sœur de Camille », même s'il n'a jamais rencontré une telle structure auparavant. En revanche, un transformeur ne sera capable de construire le sens que d'expressions dont la structure est apparue dans les données d'apprentissage. Ainsi la construction du sens dans les transformeurs repose-t-elle sur des mécanismes au moins partiellement différents de ceux mis en jeu dans les modélisations linguistiques et que l'on suppose être au cœur de la compréhension du langage par le cerveau humain.

L'association d'une architecture neuronale particulière (le transformeur) au principe de l'auto-supervision a donc conféré aux ordinateurs la faculté de *comprendre* un énoncé en langage naturel. Cette même combinaison a servi de fondation aux giga-modèles et à leurs capacités encore plus surprenantes.

4

LES SUPRENANTS POUVOIR DES TRANSFORMEURS

Les transformeurs sont le moteur de la révolution récente en TAL et en IA. Introduits pour permettre l'apprentissage de systèmes de traduction plus rapides, capables de traiter de grands corpus, ils ont été rapidement utilisés pour d'autres types de tâches. Grâce à leurs nombreuses et surprenantes capacités qui émergent spontanément des données, ils sont à la base des giga-modèles, dont le célèbre ChatGPT, et ouvrent la voie à des ordinateurs capables de maîtriser le langage aussi bien que les humains.

L'introduction des transformeurs en 2017 a provoqué un véritable raz-de-marée dans la communauté TAL. La diffusion de modèles pré-entraînés, les fameux *modèles de langue*, dans de nombreuses langues, a fondamentalement changé la manière de travailler et les compétences nécessaires en TAL. Pendant des années, la conception d'un système de TAL passait, quelle que soit la langue, le domaine ou la tâche considérée, par deux longues phases : l'annotation des données et la mise au point des représentations les plus pertinentes pour résoudre la tâche donnée.

L'utilisation de modèles pré-entraînés a grandement simplifié cette dernière étape. En effet, ces modèles permettent d'obtenir facilement (littéralement en deux lignes de code) des représentations génériques de textes, dont les performances dans une grande variété de tâches s'avèrent excellentes. Le rôle et l'intérêt des modèles pré-entraînés ont rapidement été renforcés lorsque les chercheurs ont découvert qu'il était possible de les

affiner (on parle en anglais de *fine-tuning*), c'est-à-dire de les spécialiser pour certains types de données.

De nombreuses expériences ont montré qu'il suffisait pour cela de constituer un corpus avec des données du domaine souhaité (par exemple, des textes médicaux ou juridiques) et de poursuivre l'apprentissage du modèle sur celles-ci : le modèle va successivement considérer les différentes phrases de ce nouveau corpus, prédire les mots masqués et mettre à jour ses poids pour corriger d'éventuelles erreurs. La seule différence par rapport à un apprentissage *classique* est que les poids ne sont pas initialisés de manière aléatoire, mais prennent les valeurs découvertes après l'apprentissage sur un grand corpus de données non spécifiques. Intuitivement, cette approche permet de commencer par acquérir des connaissances génériques sur la langue (par exemple, sur sa syntaxe) et de les spécialiser ensuite sur un domaine spécifique. L'avantage de ce tir en deux coups est de réduire la quantité de données nécessaires à l'apprentissage du modèle, l'essentiel des paramètres ayant déjà été appris sur le corpus générique.

L'affinage permet également de spécialiser un modèle pour une tâche spécifique en remplaçant la tête de classification (la partie du réseau qui transforme les représentations construites par l'encodeur en mots) par un autre réseau de neurones capable de résoudre une tâche différente. On peut, par exemple, adapter une nouvelle tête de classification capable de prédire l'étiquette morpho-syntaxique de chaque mot ou la polarité d'une phrase, comme nous l'avons vu au [chapitre 2](#). Cela revient simplement à apprendre une nouvelle matrice de paramètres calculant un vecteur de scores adapté à la tâche considérée à partir des représentations construites par l'encodeur. Comme dans le cas évoqué dans le paragraphe précédent, il suffit alors de continuer l'apprentissage pour à la fois affiner les représentations et apprendre la partie du modèle spécifique à la tâche considérée. Cette approche présente un intérêt majeur : étant donné que la quasi-totalité des poids du modèle a déjà une valeur permettant

d'obtenir une *bonne* représentation des mots, seul l'apprentissage des paramètres de la nouvelle tête sera nécessaire, ce qui nécessite beaucoup moins de données que s'il avait également fallu apprendre la représentation des mots.

Les transformeurs et les modèles pré-entraînés ont donc grandement simplifié le développement de systèmes de TAL, d'autant plus qu'il n'a pas fallu attendre longtemps pour que des bibliothèques (les briques composant les programmes informatiques) soient développées afin de permettre à n'importe quel programmeur de télécharger, affiner et utiliser des modèles pré-entraînés en quelques instructions triviales, sans nécessairement être un spécialiste du TAL.

Mais ils ont fait bien plus que cela ! Il est très vite apparu que ces modèles ouvraient tout un éventail de nouvelles possibilités allant bien au-delà de ce que l'on pouvait imaginer il y a seulement quelques années, et dont l'impact dépassait, de loin, le cadre du TAL.

Générer du texte

Jusqu'à présent, nous avons principalement parlé des transformeurs comme d'un outil permettant à l'ordinateur de *comprendre* un texte, c'est-à-dire de le représenter de manière appropriée pour effectuer des prédictions (comme déterminer si un avis est positif) et des inférences (par exemple, trouver des synonymes de mots ou traduire une phrase). Les transformeurs peuvent également être utilisés pour générer du texte, c'est-à-dire, en quelque sorte, pour *parler*. L'architecture d'un décodeur, le nom donné à un réseau de neurones capable de générer du texte, repose sur les mêmes composantes que l'encodeur du chapitre précédent, ainsi que sur la formulation du problème de génération comme un problème de classification (voir [ici](#)).

L'idée centrale de cette approche est de générer une phrase de manière incrémentale, mot par mot. Supposons que le réseau ait déjà généré cinq mots formant le préfixe « <bos> la musique creuse le ». Le mot « <bos> » (pour beginning of sentence – début de phrase) indique le début de la phrase et sert à amorcer le processus de génération : le premier mot est généré en considérant un préfixe ne contenant que ce mot.

L'astuce pour (ré)utiliser une architecture transformeur dans un décodeur consiste simplement à considérer que le préfixe est suivi par un mot masqué. Il devient alors possible de prédire ce mot, exactement comme un encodeur le fait lors de son apprentissage. Le décodeur commence par construire les représentations des mots du préfixe et du mot masqué en utilisant une succession de couches d'auto-attention et de réseaux à propagation avant (la même architecture qu'un encodeur). Ensuite, une tête de classification prédit le mot masqué à partir de la représentation construite de celui-ci.

Le mot généré est alors ajouté au préfixe, et le processus est répété (avec un préfixe de plus en plus long) pour générer le mot suivant, jusqu'à ce que le décodeur génère un mot spécial indiquant la fin de la phrase. Cette architecture, qualifiée d'*auto-régressive* pour souligner l'idée que la prédiction effectuée à une étape donnée est réutilisée comme entrée à l'étape suivante, est résumée sur la figure ci-dessous.

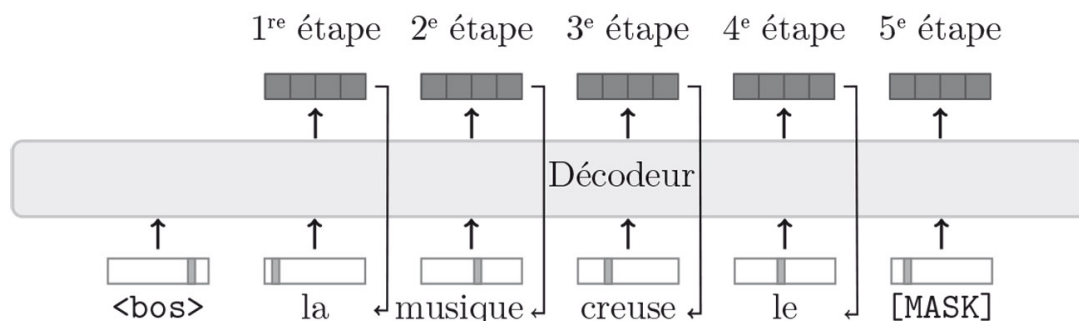


Figure 14. Principe d'un décodeur reposant sur une architecture transformeur. Le décodeur est une architecture auto-régressive :

les mots générés lors des 4 premières étapes sont ajoutés au préfixe pour permettre la génération du 5^e mot.

Le principe que nous venons de décrire peut être utilisé à la fois pour permettre à l'ordinateur de *parler* (en amorçant le processus de génération avec le mot <bos>), mais également pour *répondre*. Dans ce cas, le processus est initié en fournissant un préfixe contenant la question et en demandant au réseau de prédire les mots qui vont suivre. Le préfixe est alors généralement appelé *amorce* (en anglais *prompt*). Notons que cette amorce peut être plus ou moins longue : si la question porte sur un document, l'amorce contiendra la totalité de celui-ci suivi par la question. La taille de l'amorce qui peut être considérée est aujourd'hui l'une des principales caractéristiques des giga-modèles.

Un décodeur peut être appris à partir de textes bruts en masquant les mots d'une phrase et en retrouvant ceux-ci, exactement comme lors de l'apprentissage d'un encodeur. Il n'y a qu'une seule différence dans l'apprentissage des deux modèles : comme un décodeur ne construit que des représentations du préfixe (il ne connaît pas les mots qui vont suivre le mot masqué), plutôt que masquer des mots au hasard dans la phrase, les exemples d'apprentissage seront construits en générant tous les préfixes d'une phrase et en demandant au modèle de prédire le mot suivant, comme dans l'exemple suivant :

la musique creuse le ciel.

↓

préfixe	mot à prédire
<bos>	la
<bos> la	musique
<bos> la musique	creuse
<bos> la musique creuse	le
<bos> la musique creuse le	ciel
<bos> la musique creuse le ciel	.
<bos> la musique creuse le ciel.	<eos>

Les décodeurs ont été conçus (mais cette idée initiale sera rapidement remise en cause) pour être utilisés en conjonction avec un encodeur : ce dernier construit une représentation de l'entrée (la phrase source), qui est ensuite transférée au décodeur qui va à son tour générer, en sortie, une version transformée de l'entrée (la phrase cible). Cette architecture, appelée *encodeur-décodeur*, peut être utilisée à la fois dans des contextes monolingues dans lesquels la langue des phrases sources et des phrases cibles est la même (par exemple pour résumer ou simplifier un texte), mais aussi dans des contextes bilingues correspondant au cas où les phrases sources et les phrases cibles sont exprimées dans des langues différentes (par exemple, pour la traduction automatique).

L'information peut facilement être transférée entre le décodeur et l'encodeur en *adaptant* le mécanisme d'auto-attention. Rappelons-le, ce mécanisme, au cœur des transformeurs, permet au modèle d'apprendre quels éléments d'une phrase sont importants pour déterminer le sens et les propriétés grammaticales d'un mot donné, qui doivent donc être pris en compte dans la construction de sa représentation. Il suffit, pour cela, d'utiliser le même principe, mais cette fois en faisant porter l'attention sur les mots de la phrase source (enfin, sur leurs représentations) : ce mécanisme *d'attention croisée* entre la phrase considérée par l'encodeur et la phrase générée par le décodeur détermine pour chaque mot que prédit le décodeur quels sont les mots de la phrase source à prendre en compte. C'est, en pratique, exactement le même mécanisme que celui qui est mis à l'œuvre dans l'auto-attention (entre les mots d'une même phrase, que ce soit du côté encodeur ou du côté décodeur) et l'attention croisée (entre les mots de l'encodeur et ceux du décodeur), la seule différence porte sur les mots considérés.

Les représentations construites par le décodeur reposent ainsi sur deux sources d'information nécessaires pour générer un texte correct. Une première source d'information est constituée par les

mots du préfixe qui ont déjà été produits par le décodeur. Ces informations permettent, entre autres, de garantir que la phrase est grammaticalement correcte et respecte par exemple les règles d'accord. Elles sont complétées par des informations provenant des mots de la phrase source ou, pour être plus précis, la représentation que le décodeur a construite de ceux-ci. Ces dernières vont *contrôler* le contenu généré par le décodeur en garantissant par exemple que la phrase source et la phrase cible ont le même sens.

Deux stratégies peuvent être utilisées pour apprendre une architecture de type encodeur-décodeur qui nécessite toutes deux un corpus parallèle contenant des exemples de phrases sources et des phrases cibles associées. La première stratégie consiste à réaliser un apprentissage complet à partir du corpus, au cours duquel l'ensemble des paramètres du réseau (aussi bien les auto-attentions et les réseaux à propagation avant de l'encodeur et du décodeur que l'attention croisée) sont déterminés.

La seconde stratégie consiste à commencer par apprendre l'encodeur et le décodeur séparément sur des corpus annotés de la langue source et de la langue cible et, dans un second temps seulement, de considérer le corpus pour adapter les représentations qui auront été apprises, et surtout apprendre les paramètres de l'attention croisée. Cette dernière stratégie a l'avantage de nécessiter beaucoup moins de données parallèles (généralement difficiles à collecter), une partie des paramètres étant appris sur des données monolingues, beaucoup plus faciles à collecter.

Comme nous l'avons déjà mentionné, les transformeurs ont été introduits dans le contexte de la traduction automatique et le fameux article *Attention is all you need* (voir [ici](#)) a décrit à la fois l'architecture de l'encodeur, celle du décodeur, ainsi que le couplage entre ces deux parties à l'aide du mécanisme d'attention croisée. Mais un décodeur peut également être utilisé seul comme

un modèle génératif capable d'analyser (de comprendre !) un préfixe (constitué d'une ou de plusieurs phrases) et de le compléter en prédisant, mot après mot, la ou les phrases qui vont suivre.

Les modèles génératifs ouvrent de nouvelles possibilités d'application des transformeurs allant au-delà de la simple génération de texte. Ils peuvent également être utilisés pour répondre à des questions : pour connaître la capitale de la France, il suffit de demander à un modèle génératif de compléter le préfixe « La capitale de la France est... » Le mot prédit par le modèle sera alors vraisemblablement la réponse. Nous reviendrons en détail sur cette utilisation, mais il est important de bien comprendre dès maintenant que la réponse fournie par le modèle de langue repose uniquement sur les connaissances stockées dans ses paramètres et extraites des données d'apprentissage. Comme nous l'avons mentionné précédemment, ces connaissances sont principalement axées sur les relations entre les symboles. La réponse donnée est donc fondamentalement différente de celle d'un moteur de recherche, qui va retrouver sur l'Internet un document permettant à une personne d'extraire des informations pertinentes, d'identifier l'auteur de la page et de jauger son niveau de fiabilité.

L'idée de construire un modèle ne reposant que sur un décodeur a été mise en œuvre dès 2018 (l'année suivant l'introduction des transformeurs) par OpenAI. L'entreprise diffusa un premier modèle pré-entraîné de ce type qui peut être vu comme le pendant de BERT (constitué uniquement d'un encodeur) sous le nom de « transformeur génératif pré-entraîné » soit, en anglais *generative pre-trained transformer*, le fameux GPT. La révolution ChatGPT était en marche !

Finalement, il existe donc deux familles de modèles reposant sur l'architecture transformeur : les modèles constitués uniquement d'un encodeur, dont le représentant le plus célèbre est BERT, et les

modèles auto-régressifs ou génératifs constitués uniquement d'un décodeur. Ces deux familles vont toutes deux présenter des capacités surprenantes.

Représenter toutes les langues

Une première capacité surprenante et inattendue des transformeurs est la possibilité de développer des *modèles multilingues*. L'idée est on ne peut plus simple : plutôt que d'apprendre un modèle uniquement sur des phrases d'une même langue (un corpus *monolingue*), le corpus d'apprentissage sera *multilingue*, c'est-à-dire constitué de phrases en plusieurs langues. Le modèle est entraîné en suivant exactement la même procédure que celle décrite dans le chapitre précédent¹ : pour chaque phrase, le modèle prédira les mots masqués et (éventuellement) adaptera ses poids pour corriger ses prédictions ; la seule différence est que la langue changera d'une phrase à l'autre.

Mélanger ainsi les données sans apporter aucune modification à la méthode d'apprentissage dote les transformeurs de nouvelles capacités étonnantes. Comme on pouvait s'y attendre, un modèle entraîné sur un mélange de phrases en français et en anglais (ou sur toute autre combinaison de langues) saura représenter des mots français et anglais de manière similaire à un modèle monolingue ou, pour être plus précis, à une juxtaposition de modèles monolingues. Mais, et c'est une surprise, la représentation d'un mot français sera très proche de la représentation de sa traduction en anglais : comme le montre la figure ci-dessous, les représentations des deux langues sont alignées, et la structure de l'espace met directement en correspondance les mots qui sont des traductions l'un de l'autre.

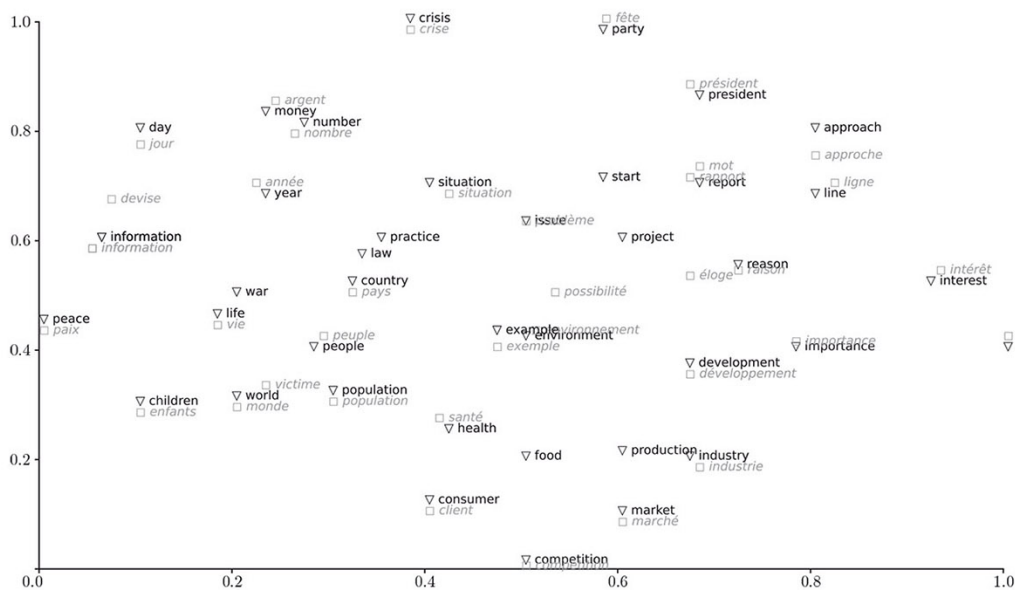


Figure 15. Projection des représentations d'un modèle bilingue anglais-français. Cette représentation montre que les mots qui sont des traductions l'un de l'autre ont des représentations proches.

Ce résultat est complètement inattendu puisque jamais le modèle n'a reçu la moindre information sur les correspondances entre langues : les données à partir desquelles il est appris ne contiennent aucune information sur la manière dont un mot ou une phrase en français se traduisent en anglais. Le modèle a fait *émerger* spontanément ces connaissances des données d'apprentissage, exactement comme il a fait émerger des notions liées à la syntaxe de la phrase. Ce résultat est d'autant plus surprenant que depuis longtemps, les chercheurs développaient des méthodes spécifiques afin d'induire automatiquement des lexiques bilingues sans jamais réussir à atteindre des résultats de cette qualité.

Il n'existe pas, à l'heure actuelle, de certitudes (de *preuves*, au sens mathématique) sur la manière dont les modèles parviennent à faire émerger des correspondances entre les langues. Plusieurs expériences ont montré qu'au moins deux facteurs (potentiellement complémentaires) semblaient être à l'œuvre. Le premier est la présence de mots communs entre les langues, notamment les noms propres, les symboles, les nombres, ou ce que les linguistiques appellent des *mots transparents* : ces mots à la

forme et au sens identiques dans les deux langues serviront en quelque sorte de point d'ancrage entre les représentations des mots anglais et celles des termes français. Ainsi, le mot « animal » aura, en français, une représentation proche de mots comme « bête », « chat » ou « lapin » et, en anglais, de mots comme « *beast* », « *cat* » ou « *rabbit* ». Comme la représentation de « animal » sera très proche voire identique en français et en anglais (au moins dans les premières couches), celle-ci va *contraindre* les représentations des mots similaires à figurer dans un voisinage proche.

Une autre explication avancée est que le modèle de langue construit des représentations abstraites des mots qui sont indépendantes de la forme de surface (la suite de caractères) : intuitivement, les différentes formes de « chat » (les variations en genre et en nombre, voire les synonymes) sont toutes associées à la même représentation abstraite. Un modèle multilingue n'a donc qu'à associer les formes de surface dans les différentes langues à ces représentations suffisamment abstraites pour être identiques d'une langue à l'autre.

La capacité d'un modèle de langue multilingue à construire des représentations très proches de mots, qui sont des traductions l'un de l'autre, a un énorme intérêt pratique : si le modèle construit exactement la même représentation pour une phrase et sa traduction, les prédictions qui en découleront, que ce soit une traduction, un résumé ou une simple tâche de classification, coïncideront. Ainsi, il n'y a, intuitivement, plus besoin de disposer de corpus étiquetés dans toutes les langues pour lesquelles on souhaite développer un système de TAL : il suffit de disposer d'annotations pour une langue donnée, et le modèle pourra être appliqué pour toutes les langues.

Cette intuition a été confirmée expérimentalement par de nombreuses expériences qui ont montré que les modèles multilingues pouvaient transférer les connaissances qu'ils avaient

appries à une autre langue. Il est donc possible de développer des systèmes de TAL sans disposer d'aucune donnée annotée dans cette langue. Certaines expériences ont même montré que les modèles multilingues permettaient, dans certains cas, d'obtenir de meilleures performances que des modèles monolingues : en apprenant simultanément à représenter les mots de plusieurs langues, un modèle découvrira de meilleures représentations.

Les modèles multilingues présentent également un intérêt pratique très important : une entreprise souhaitant intégrer un système de TAL dans son application n'aura besoin de développer, tester, valider, et surtout, de maintenir qu'un seul modèle, quelle que soit la langue de ses utilisateurs. Ce sera pour elle un gain de temps et d'énergie considérable.

En fin de compte, de nombreux modèles de langage, y compris ChatGPT, sont actuellement entraînés sur des corpus multilingues, et donc capables de représenter, selon les modèles, entre cinquante et une centaine de langues. À l'heure actuelle, il semble difficile d'entraîner des modèles capables de dépasser ce seuil. En effet, au-delà des cent langues les plus parlées, il n'existe pas de corpus de taille suffisante pour entraîner des modèles de langue. Mais les chercheurs ont également mis en évidence ce qu'ils appellent la *malédiction du multilinguisme* : si les performances des modèles multilingues s'améliorent à chaque ajout d'une nouvelle langue, à partir généralement de 100 langues, les performances commencent à chuter, pour des raisons qui ne sont pas encore entièrement comprises.

Des modèles multimodaux

Bien que les transformeurs aient été initialement conçus pour améliorer la représentation des mots, ils ont été rapidement appliqués à d'autres types de données. Ils ont ainsi été utilisés pour apprendre des représentations du code source des programmes informatiques : ces derniers sont spécifiés sous forme

d'instructions écrites dans des langages spéciaux comme Python ou Java, mais qui, finalement, ne sont que des suites de caractères. En ajoutant les millions de lignes de code que l'on peut trouver en ligne², des modèles permettant de générer du code à partir de spécifications en langage naturel ont été très rapidement diffusés. Aujourd'hui, de nombreux outils de développement incluent une IA aidant les programmeurs dans l'écriture du code informatique. Dès 2018, cette architecture a également été utilisée pour représenter des données audios et, en 2020, elle a été appliquée aux images, remplaçant avec succès, dans les deux cas, des architectures neuronales spécifiquement conçues pour ces types de données.

L'architecture transformeur adaptée à la représentation des images comporte deux étapes (qui sont en tout point identiques à celles utilisées pour représenter les données audios). Dans la première étape, l'image est découpée en *patches*, c'est-à-dire des carrés de taille fixe, et un premier réseau de neurones permet de déterminer une représentation initiale de chaque patch. Ensuite, un encodeur avec une architecture transformeur, composé d'une succession de couches d'auto-attention et de couches à propagation avant, ajoute des informations contextuelles (provenant des autres patches) à ces représentations. La figure ci-dessous synthétise le principe de ce modèle appelé *ViT* pour *VisionTransformer*.

Les paramètres du modèle sont généralement appris de manière supervisée à partir de l'un des nombreux corpus qui, comme ImageNet, associe chaque image à des étiquettes décrivant les objets représentés. Les représentations sont progressivement mises à jour pour arriver à prédire l'étiquette associée à l'image.

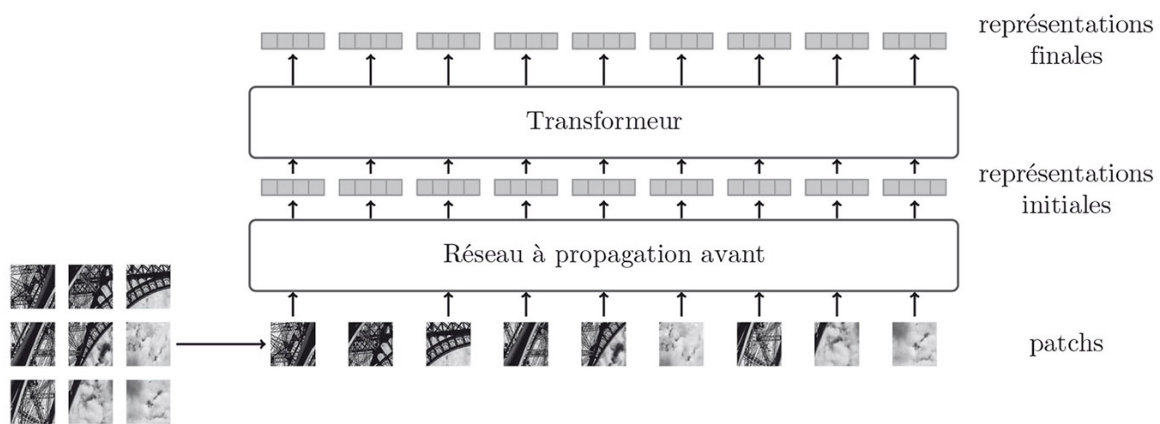


Figure 16. Principe du ViT : l'image est découpée en *patches*.
Un premier réseau de neurone construit la représentation initiale de chaque patch, représentations qui sont ensuite contextualisées à l'aide d'un transformeur.

Si une même architecture peut être utilisée pour construire des représentations vectorielles des textes et des images, il faut encore assurer que les représentations des différentes modalités soient *alignées* (mises en correspondance). Contrairement aux modèles multilingues, où cette mise en correspondance des représentations pour différentes langues se fait, de manière un peu magique, automatiquement, les modèles multimodaux requièrent une méthode d'apprentissage spécifique pour aligner ces représentations.

En 2021, les chercheurs d'OpenAI ont introduit une nouvelle approche pour apprendre des représentations alignées de textes et d'images, baptisée CLIPS pour *Contrastive Language-Image Pre-training*. Cette approche repose sur la conjonction de deux éléments : un immense corpus composé de 400 millions d'images, chacune associée à une phrase en anglais la décrivant, et un nouveau critère d'apprentissage reposant sur la capacité à choisir la bonne association plutôt que de (re)générer la partie masquée, comme pour l'entraînement des modèles de langue. C'est pourquoi cette méthode d'apprentissage est qualifiée de *contrastive* puisqu'elle repose sur la capacité du modèle à distinguer les représentations similaires de celles qui sont différentes.

Pour constituer un corpus de si grande taille (rappelons qu'ImageNet, le corpus dont la taille avait permis aux réseaux de neurones d'enregistrer leur premier succès, ne contient que 10 millions d'images et que celles-ci ne sont décrites que par des mots), les chercheurs se sont appuyés sur une trentaine de bases de données existantes, associant soit des images à leurs légendes, soit des images à des mots. Dans ce dernier cas, les mots ont été automatiquement transformés en phrases à l'aide de patrons (des modèles de phrases similaires aux patrons utilisés en couture) : une image annotée avec l'étiquette « chien » est associée, par exemple, à la phrase « photo d'un chien ». Pour compléter ces corpus, on a également collecté des images en utilisant 500 000 requêtes construites automatiquement à partir des mots et des paires de mots les plus fréquents dans la Wikipédia en anglais, conservant jusqu'à 20 000 associations entre une image et un texte pour chaque requête. Les efforts et l'ingéniosité déployés pour constituer un corpus annoté aussi grand montrent bien l'importance que les données, et surtout les données annotées, continuent de jouer, aujourd'hui encore, dans les progrès de l'IA !

Ce corpus a été utilisé pour apprendre conjointement les représentations des textes et des images en utilisant une tâche très simple, similaire à celle consistant à retrouver la partie masquée d'une phrase ou d'une image. L'idée principale est de prendre un ensemble de paires associant une image à sa description (32 768 dans la première version du modèle) et de retrouver, pour chaque image, la bonne description parmi l'ensemble des descriptions de l'ensemble considéré. Cela implique de mesurer la similarité entre tous les vecteurs représentant d'un côté les images, et de l'autre les textes (ce qui correspond quand même à un peu plus d'un milliard de comparaisons), en s'assurant que le texte le plus similaire à une image donnée correspond bien à celui associé à cette image dans le corpus d'apprentissage. Si ce n'est pas le cas, les représentations du texte et de l'image sont ajustées par rétropropagation du gradient afin de corriger l'erreur.

Exactement comme pour la tâche consistant à retrouver les mots masqués, il est surprenant (pour ne pas dire complètement déroutant) de voir qu'une tâche aussi simple puisse construire des représentations si riches qu'elles résolvent des problèmes que l'on pensait hors de portée pour un ordinateur il y a quelques années (voire quelques mois) encore.

Ces représentations conjointes des textes et des images sont au cœur d'un autre grand succès de l'IA qui a fait (et continue de faire) la une des journaux : DALL·E, le générateur d'images d'OpenAI. Ce logiciel est capable de générer, en faisant souvent preuve de créativité, des images, qu'il s'agisse de photos, de dessins ou de peintures, à partir d'une description textuelle. Les premiers exemples d'images générées par DALL·E³ démontrent clairement la créativité du programme : il est, par exemple, capable de créer des images de chaises en forme d'avocat ou d'escargots en forme de harpe.

Pour produire des images, DALL·E emploie simplement un décodeur (qui repose évidemment sur un transformeur) pour générer une image à partir d'une représentation vectorielle fonctionnant sur un principe proche de celui des décodeurs qui génèrent du texte (comme ceux décrits au début de ce chapitre). La principale différence est que celui-ci est appris selon l'algorithme de la *diffusion stable*, qui permet d'obtenir plus d'originalité dans les images générées.

La possibilité de produire des images à partir d'une simple description textuelle a de quoi faire rêver. Effectivement, les journalistes de la presse grand public se sont très vite emparés du sujet. Les articles vantaient tantôt les mérites (ou les défauts) de DALL·E et de ses concurrents comme MidJourney ou Imagen (la réponse d'Alphabet à DALL·E), tantôt s'interrogeaient sur l'impact que ces modèles allaient avoir sur notre quotidien (nous reviendrons sur cette question dans le [chapitre 5](#)). Et pourtant, les transformeurs sont encore loin d'avoir révélé tout leur potentiel⁴.

L'efficacité déraisonnable des données

La course au gigantisme

La capacité la plus surprenante et inattendue des transformeurs est apparue lorsque les chercheurs ont entraîné des modèles de plus en plus grands. La diffusion de BERT, en 2019, a marqué le début d'une course aux armements qui est toujours en cours. L'objectif ? Apprendre un modèle avec le plus grand nombre de paramètres possibles.

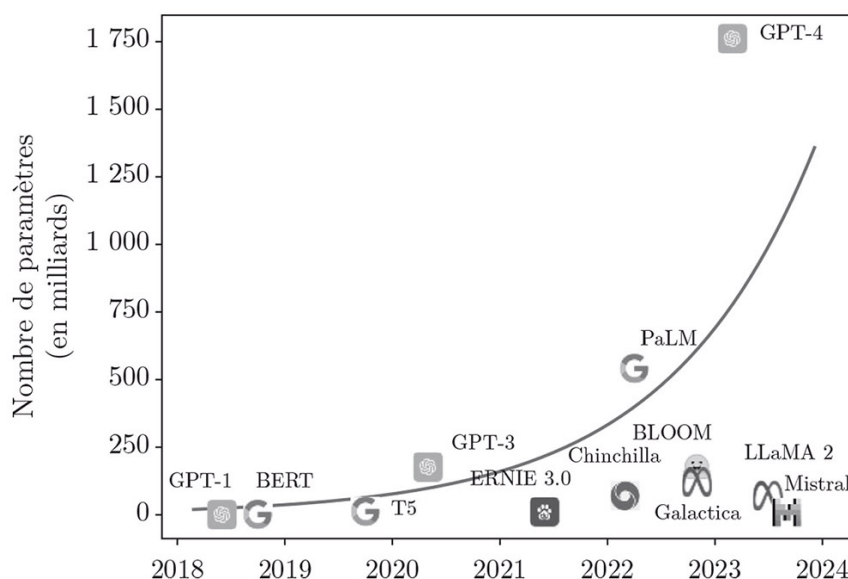


Figure 17. Évolution du nombre de paramètres dans les modèles de langue.

Comme nous l'avons vu, un modèle transformeur est composé d'une succession de couches, qui construisent progressivement une représentation de plus en plus abstraite de la phrase. Chaque couche possède ses propres paramètres, et en augmentant la profondeur du modèle via l'ajout de couches, le nombre de paramètres du modèle augmente mécaniquement, renforçant ainsi sa capacité à modéliser des relations de plus en plus complexes entre les mots.

Cette course au gigantisme est motivée par un constat simple : il est rapidement devenu évident que l'augmentation du nombre de paramètres fait *émerger* de nouvelles capacités dans les modèles.

C'est l'aspect presque magique de la taille : le modèle s'enrichit de capacités nouvelles à mesure que son nombre de paramètres et la taille de son corpus d'apprentissage augmentent. Très vite, les modèles sont devenus si grands qu'ils ont été qualifiés de *large language models* (LLM) ou, en français, de giga-modèles.

Cependant, cette augmentation a un coût, tant en termes de puissance de calcul que de données requises pour entraîner le modèle et trouver la valeur optimale des paramètres. Et ce n'est pas sans raison que, comme le montre clairement la [figure 17](#) (page précédente), tous les modèles de langue ou presque ont été développés par de grandes entreprises. En effet, plus un modèle est profond, plus la quantité de calculs (en pratique : le nombre de matrices à multiplier) nécessaire pour prédire un mot ou mettre à jour les paramètres augmente significativement. De plus, un corpus d'apprentissage plus grand est nécessaire pour permettre au modèle de découvrir des relations pertinentes.

Par conséquent, pour participer à la course à la taille des modèles, il faut une infrastructure informatique conséquente dotée, entre autres, d'un grand nombre de GPUs, d'un réseau efficace pour le transfert de données entre les différentes machines, ainsi que des capacités de refroidissement adéquates, les GPUs dissipant beaucoup de chaleur lorsqu'ils fonctionnent. Au-delà de l'investissement matériel (un seul GPU coûte, rappelons-le, plusieurs milliers d'euros), on a besoin d'ingénieurs spécialisés capables de s'assurer du bon fonctionnement de l'ensemble et de garantir que les calculs se répartissent efficacement sur les différents GPUs (et CPUs) disponibles, sans provoquer de goulots d'étranglement. Tous ces problèmes techniques sont particulièrement complexes et nécessitent une expertise solide.

Les modèles GPT

Pour bien comprendre l'intérêt de cette course et la taille des modèles mis en jeu, le plus simple est d'examiner l'évolution de

l'un des modèles de langage les plus emblématiques : GPT. Comme nous l'avons expliqué au début de ce chapitre, GPT est un modèle de langue – développé par OpenAI – basé sur un décodeur utilisant l'architecture transformeur.

La première version du modèle, nommée GPT-1, a été publiée en juin 2018, soit un an après l'introduction des transformeurs et quelques mois avant la diffusion de BERT. Il s'agissait d'un *petit* modèle de 120 millions de paramètres (trois fois moins que BERT), comprenant 12 couches de décodeur, chacune avec 12 têtes d'attention, et utilisant des représentations cachées des mots de dimension 3 072. Le modèle a été entraîné sur le même corpus de 7 000 livres que BERT. Bien qu'il ait obtenu de bonnes performances sur de nombreux tests, GPT-1 revêtait surtout un intérêt technique : il a montré la possibilité d'apprendre un modèle de langue reposant uniquement sur un décodeur et donc capable de générer du texte, ouvrant ainsi la voie aux modèles dits *génératifs* et, plus généralement, à l'IA générative. La diffusion du modèle s'est cantonnée à un cercle restreint de chercheurs.

Un peu plus d'un an plus tard (en février 2019), OpenAI a annoncé la deuxième version de son modèle de langue, GPT-2. Ce modèle était dix fois plus volumineux, avec près de 1,5 milliard de paramètres. Cette fois, le modèle comprenait 48 couches, chacune avec 25 têtes d'attention, et les représentations des mots étaient de dimension 1 600. Il a été entraîné sur 40 Go de textes, principalement issus de 8 millions de pages web soigneusement sélectionnées : l'essentiel des pages considérées étaient des pages citées dans des messages sur Reddit⁵ ayant reçu de bonnes notes, un moyen simple d'évaluer la *qualité* de leur contenu. Il n'y a pas que la quantité des données qui est importante, mais également leur qualité ! Mis à part quelques détails techniques, GPT-2 était parfaitement identique à GPT-1 à l'exception du nombre de paramètres et de la taille du corpus d'apprentissage. Pour être clair, il n'y avait, du point de vue scientifique, aucune découverte

ou avancée supplémentaire entre ces deux modèles, du moins du côté de l'IA ou du TAL.

Pourtant, GPT-2 a montré d'excellentes performances sur de nombreuses tâches, surpassant même parfois BERT. Mais le plus surprenant était la capacité du modèle à générer des textes syntaxiquement corrects, sémantiquement cohérents et écrits dans un style similaire à l'amorce qui lui était fournie et ce dans plusieurs langues (GPT-2 étant un modèle multilingue). GPT-2 a ainsi pu être utilisé pour terminer des chansons et des poèmes, mais aussi pour écrire des articles de journaux à partir de leur titre en prenant l'initiative d'inclure des citations (fausses mais insérées à des endroits pertinents) et en inventant les statistiques dont il avait besoin⁶.

La qualité des textes était telle qu'OpenAI a jugé, à juste titre, comme nous en discuterons dans le chapitre 5 (voir [ici](#)), que GPT-2 pourrait être dangereux. Il pouvait, en effet, être utilisé pour générer de fausses pages web, de faux messages dans le cadre de campagnes de désinformation, ou plus simplement, pour rédiger de faux avis sur des sites de vente en ligne. Dans les deux cas, la qualité des textes suffisait à donner l'impression qu'ils avaient été écrits par un humain. À cet égard, le titre de l'article présentant le modèle dans le journal *Le Monde* est sans équivoque : « Fausses informations : un programme d'IA jugé trop dangereux pour être rendu public »⁷.

Le choix d'OpenAI, qui n'était encore qu'une organisation à but non lucratif, de présenter son modèle comme une arme et de ne le rendre initialement accessible qu'à certains journalistes choisis avec soin, a marqué un tournant dans le développement des modèles de langage, en plus de créer le buzz. Alors que la recherche se menait au grand jour, les entreprises ou laboratoires partageant toutes les informations nécessaires pour permettre à tous les acteurs de reproduire leurs expériences et de valider leurs modèles, OpenAI a été très mutique sur les données ayant servi à

entraîner GPT-2, et il a fallu de nombreuses pressions, notamment de la communauté scientifique, pour que l'entreprise accepte de diffuser son modèle pré-entraîné et que les autres chercheurs puissent l'utiliser dans leurs systèmes et en étudier ses étonnantes capacités.

La troisième version de GPT, GPT-3, a été publiée en mai 2020. Une fois encore, OpenAI se contentait d'augmenter la taille du modèle précédent, sans rien changer à son architecture. Mais quelle inflation de chiffres ! Elle incluait 175 milliards de paramètres, soit 100 fois plus que GPT-2. Elle comprenait 96 couches, chacune ayant 96 têtes d'attention, et utilisait des vecteurs de dimension 12 288 pour représenter les mots. Le corpus d'apprentissage était également sans précédent, avec près de 500 milliards de mots provenant principalement de sites web toujours sélectionnés avec soin (82 % des données), ainsi que de Wikipédia et de corpus de livres. Ses capacités, sur lesquelles nous reviendrons bientôt en détail (voir [ici](#)), sont complètement stupéfiantes, au point que beaucoup considèrent que ce modèle a entraîné un changement de paradigme en TAL, voire dans toute l'IA.

Cette fois-ci, ni les données, ni le code, ni le modèle n'ont été partagés publiquement. Il n'est désormais plus possible de télécharger le modèle et de l'exécuter sur son ordinateur : celui-ci fonctionne seulement sur des serveurs contrôlés par OpenAI, auxquels les données sont envoyées pour obtenir des prédictions. Le service est également devenu payant : OpenAI facture chaque mot échangé, que ce soit dans le préfixe envoyé au serveur ou dans la réponse générée par le modèle. Au moment où ces lignes sont écrites (janvier 2024), il faut déboursier jusqu'à 10 centimes pour envoyer 7 500 mots et 20 centimes chaque fois que le modèle aura généré 7 500 mots.

Ce changement dans le mode de fonctionnement, qui a des implications fortes (que nous aborderons dans le [chapitre 5](#)),

s'accompagne de deux virages stratégiques majeurs : OpenAI, qui était à l'origine une société à but non lucratif (comme le suggère l'adjectif *open* composant son nom) dont le seul objectif était de développer une IA générale, est devenue une entreprise à « but lucratif plafonné » afin d'attirer des investisseurs. Elle a également établi un partenariat avec Microsoft en juillet 2019. L'éditeur du célèbre système d'exploitation Windows et de la non moins célèbre suite bureautique Office a investi 1 milliard de dollars dans la société, devenant ainsi un des principaux acteurs de la recherche en IA. L'entreprise a surtout fourni à OpenAI un accès privilégié à Azure, sa plateforme de *cloud computing*, qui offre, entre autres, une infrastructure de calcul (aussi bien sur CPU que sur GPU) très performante nécessaire à l'apprentissage de modèles toujours plus grands.

OpenAI a publié la dernière version de son modèle, GPT-4, en mars 2023 (après la sortie de ChatGPT). Selon l'avis général de ses utilisateurs, GPT-4 est encore plus performant que ses prédécesseurs. Le modèle est notamment multimodal et accepte, en entrée, aussi bien des textes que des images. Cependant, le secret qui entoure ce modèle et son accès strict, uniquement par une API payante encore plus coûteuse que celle de GPT-3, ont rendu impossible toute évaluation précise par la communauté scientifique. En effet, aucune information n'a été communiquée concernant le modèle lors de sa diffusion : ni sa taille, ni les données utilisées pour l'apprentissage, ni l'architecture du modèle n'ont été rendus publics. Il a fallu attendre novembre 2023 pour que les premières informations sur le modèle commencent à fuiter.⁸

Il semble que cette fois-ci, OpenAI n'ait pas simplement augmenté la taille de son modèle (bien qu'il compte finalement 10 fois plus de paramètres que GPT-3). En effet, GPT-4 n'est plus un modèle monolithique de grande taille, mais plutôt une *réunion* de 8 modèles, comportant chacun 220 milliards de paramètres. Chaque modèle est entraîné sur des tâches et des données

différentes⁹, et est, en quelque sorte, spécialisé dans un type de question spécifique. GPT-4 combine toutes les réponses pour produire sa réponse finale, réalisant ainsi un « mélange d'experts » (*mixture of experts*). L'idée de fusionner différents modèles est aussi vieille que l'apprentissage statistique lui-même et a été remise au goût du jour par des chercheurs de Google deux ans plus tôt¹⁰. Une fois de plus, OpenAI se démarque davantage par sa maîtrise technique et son marketing que par sa capacité à innover.

En gardant secrètes toutes les informations techniques liées à son modèle, OpenAI a certainement voulu compliquer la tâche de ses concurrents de plus en plus nombreux. Mais le mystère entourant GPT-4 a également (et surtout) alimenté les spéculations sur les capacités de ce nouveau modèle, donnant l'impression qu'OpenAI avait une fois de plus réalisé une avancée majeure en IA, se rapprochant un peu plus de ce que l'on appelle une intelligence artificielle générale, c'est-à-dire une IA ayant une compréhension et une capacité cognitives similaires à celles d'un être humain. Une belle manière de créer un buzz, tout en masquant le fait que les progrès d'OpenAI reposent avant tout sur de vieilles méthodes remises au goût du jour par ses concurrents et sa capacité à mobiliser suffisamment de ressources pour entraîner des modèles gigantesques. Il ne faut pas oublier que le développement de l'IA est devenu une guerre économique !

Du côté des concurrents

La famille des GPT est très certainement le modèle de langue le plus connu et le plus visible, en grande partie parce qu'OpenAI a toujours été à la pointe de la recherche sur cette question et a souvent eu une longueur d'avance sur ses concurrents. De plus, l'entreprise a bénéficié d'une forte visibilité grâce à ses spectaculaires démonstrations.

Cependant, de nombreuses autres entreprises ont développé leurs propres modèles de langue, bien sûr les GAFAM et leurs homologues chinois, mais plusieurs startup se sont également lancées dans la course. Par exemple, Google a annoncé la création de trois giga-modèles : Chinchilla (mars 2022, 70 milliards de paramètres, mais entraîné sur 4 fois plus de données), qui obtient de meilleures performances que GPT-3 sur de nombreuses tâches, LaMDA (janvier 2022, 137 milliards de paramètres) et PaLM (avril 2022, 540 milliards de paramètres). Malheureusement, tous ces modèles sont fermés et ne sont pas directement accessibles au grand public.

En revanche, Meta continue de laisser libre accès à ses principaux modèles de langue, notamment LLaMA, un modèle de 65 milliards de paramètres diffusé en février 2023, et LLaMA2 (70 milliards de paramètres entraînés sur 2 000 milliards de mots) diffusé en juillet de la même année.

En parallèle des modèles développés par de grandes entreprises, plusieurs projets ont été lancés pour créer des modèles plus ouverts. Par exemple, la startup franco-américaine Hugging Face, qui développe un ensemble d'outils (ce que les informaticiens appellent une *bibliothèque*) particulièrement populaires pour utiliser et affiner des modèles pré-entraînés, a initié le projet Big Science qui a réuni entre mai 2021 et mai 2022 un consortium composé de plus de 1 000 chercheurs issus de 250 laboratoires, et dont le fruit est un modèle de langue ouvert et gratuit, BLOOM.

BLOOM intègre 176 milliards de paramètres et a été entraîné sur un corpus de 1,4 téraoctet de textes sélectionnés pour leur « fiabilité ». Son entraînement a nécessité 5 millions d'heures de calcul, mobilisant des centaines de GPU en parallèle pendant 11 semaines. Ce modèle est toutefois moins performant que GPT-3 (bien qu'il soit plus gros), une belle preuve de l'expertise accumulée par OpenAI.

Ainsi, si nous parlerons essentiellement de GPT-3, le giga-modèle sur lequel s'appuie ChatGPT, dans la suite de ce chapitre, il ne faut pas perdre de vue que le paysage des giga-modèles est bien plus divers et que de nombreux autres modèles offrent des caractéristiques similaires à celle du modèle d'OpenAI.

Un nouveau paradigme pour le TAL

Ce n'est pas tant à cause de sa taille que GPT-3 a marqué les esprits que par les nouvelles capacités qu'il offrait par rapport à GPT-1 et GPT-2 (alors que seule sa taille le distinguait de ces deux modèles !). En plus de générer des textes syntaxiquement, sémantiquement et stylistiquement cohérents, GPT-3 pouvait en effet accomplir des tâches à partir d'une simple amorce (*prompt*), c'est-à-dire une description en langage naturel accompagnée ou non de quelques exemples d'entrées et de sorties (on parle plutôt ici de 2 ou 3 exemples, et non des corpus étiquetés nécessaires à l'apprentissage supervisé). Le modèle va simplement *lire* l'amorce et prédire les mots suivants qui correspondront à sa réponse.

Prenons un exemple, un peu artificiel il est vrai, de la réalisation d'opérations arithmétiques. GPT-3 est tout à fait capable d'additionner et de soustraire des nombres à deux chiffres, voire dans une certaine mesure, à trois chiffres. Il suffit de lui fournir une amorce, telle que :

Q: Combien font 48 plus 76? A: 124

Q: Combien font 84 plus 67? A: 151

Q: Combien font 4 plus 76? A: [MASK]

Il faut ensuite demander au modèle de prédire le mot masqué (la tâche pour laquelle il a été entraîné) pour que celui-ci réalise le calcul demandé et réponde 80 après n'avoir eu accès qu'à deux exemples annotés (les deux premières lignes de l'amorce).

Si l'émerveillement des chercheurs devant cette capacité peut sembler incompréhensible, étant donné la trivialité des opérations en question, il ne faut pas oublier que, contrairement à une calculatrice, GPT-3 n'a jamais été programmé pour effectuer des opérations arithmétiques. Il a réussi à comprendre, sans aucune intervention humaine, le sens à donner à ces opérations et la manière de les exécuter, en se basant uniquement sur de vastes quantités de texte et en apprenant à résoudre une tâche sans rapport avec l'arithmétique et très simple (prédire le mot suivant).

Ce qui est encore plus surprenant, c'est que la capacité à additionner ou soustraire des nombres émerge soudainement lorsque la taille du modèle atteint un certain seuil. Cela n'est pas sans rappeler les transitions de phase en physique : à partir d'une certaine taille, le modèle semble tout à coup capable d'effectuer des opérations parfaitement, ou presque, alors qu'il était incapable de le faire jusqu'à ce seuil.

La capacité de GPT-3 à accomplir des tâches, comme l'exemple des opérations arithmétiques que nous venons de décrire, pour lesquelles il n'a été ni programmé ni entraîné, a donné naissance à une nouvelle approche du développement de systèmes de TAL appelée *zero-shot learning* lorsque seule une amorce est fournie, et *few-shot learning* lorsqu'un petit nombre d'exemples sont également fournis. Cette capacité apporte ainsi une réponse à une limite majeure de toutes les approches antérieures du TAL : le besoin de disposer de données annotées en quantité suffisante pour la tâche, la langue et le domaine considéré.

Rapidement, de nombreuses amorces différentes ont été proposées et testées. Par exemple, plutôt que d'utiliser un système de traduction appris à partir d'un corpus parallèle, il est possible d'utiliser l'amorce suivante :

[Anglais] : No nightingale did ever chaunt

[Français] : Jamais rossignol ne chanta

[Anglais] : More welcome notes to weary hands

[Français] : Si doucement pour le repos
[Anglais] : Of travelers in some shady haunt,
[Français] : De voyageurs en un coin d'ombre
[Anglais] : Love is not love which alters when
it alteration finds
[Français] : [MASK]

Avec cette seule amorce, on peut obtenir un système capable de traduire (plus ou moins correctement) presque n'importe quelle paire de langues. Cette amorce contient deux parties : une série de phrases traduites avec l'indication pour chaque phrase de la langue utilisée (les trois premières paires) ; la dernière paire, incomplète, contient la phrase à traduire, l'indication de la langue cible étant suffisante pour *déclencher* la traduction. Une étude détaillée menée par des chercheurs d'Alphabet¹¹ montre que les traductions ainsi obtenues sont moins bonnes que celles obtenues par un système de traduction *traditionnel*, mais demeurent de très bonne qualité. C'est un résultat remarquable, compte tenu du fait qu'un système traditionnel est appris à partir de millions de traductions, alors qu'il suffit ici de fournir quelques exemples dans l'amorce.

Des approches basées sur des amorces ont été utilisées pour résoudre tous les problèmes traditionnels du TAL : rédaction de résumés, simplification de texte, vérification orthographique, amélioration de la lisibilité, extraction d'informations, etc. Du jour au lendemain ou presque, de nombreuses entreprises (ainsi que des chercheurs) qui avaient développé des solutions spécifiques ont dû revoir leurs objectifs, voire leur modèle économique, tant le risque que tout le code qu'elles avaient écrit puisse être remplacé par quelques mots en langage naturel et un appel (payant) au modèle d'OpenAI. Il était devenu littéralement possible de développer des systèmes de TAL en quelques heures, et une nouvelle spécialité, appelée *prompt engineering*, a même commencé à apparaître et à figurer dans les offres d'emploi.

GPT-3 (et les autres giga-modèles) a donc complètement révolutionné le TAL et, plus généralement, l'IA. Jusqu'à présent, toutes les méthodes que nous avons vues reposaient sur l'idée qu'il était nécessaire d'entraîner, à partir d'un corpus de données annotées spécifiques, un modèle pour chaque tâche, et que celui-ci ne pourrait accomplir que cette tâche particulière. GPT-3 a été le premier modèle *générique* capable de traiter un grand nombre de tâches (pour ne pas dire *toutes*), et ce sans nécessiter d'entraînement spécifique ni de corpus d'apprentissage, en se basant uniquement sur une description de la tâche et/ou quelques exemples. Les capacités de GPT-3 vont donc bien au-delà de la capacité des transformeurs à comprendre des phrases et à en construire des représentations abstraites que nous avons évoquées au chapitre 3 (voir [ici](#)).

Cependant, il est important de se rappeler que les performances obtenues par GPT-3 sont généralement inférieures à celles d'un système spécifique. De plus, le coût lié à l'utilisation de GPT-3, que ce soit en termes de temps de calcul ou de puissance informatique requise pour faire fonctionner le modèle, est considérablement plus élevé que celui d'un modèle spécifique.

Après GPT-3 ? La route vers ChatGPT

Après la sortie de GPT-3, OpenAI a poursuivi ses efforts pour améliorer les performances de son modèle phare. En 2022, l'entreprise a lancé InstructGPT, un nouveau giga-modèle. Contrairement aux évolutions précédentes, InstructGPT n'est pas plus grand que GPT-3. Mais il a l'avantage d'intégrer deux nouvelles idées qui ont considérablement amélioré ses capacités à comprendre et à parler : l'affinage sur des corpus d'instructions et l'alignement avec les intentions des utilisateurs.

Apprendre à répondre aux instructions

La première, appelée *instructions fine-tuning*, consiste à affiner un giga-modèle en suivant le principe exposé au début de ce chapitre (voir [ici](#)). L'opération se base sur des corpus contenant des *instructions* qui associent des amorces à des réponses humaines. Ces corpus d'instructions ont été constitués dès la sortie de GPT-3, initialement pour analyser les amorces soumises par les utilisateurs de ce service et d'autres giga-modèles.

Plusieurs corpus d'instructions sont aujourd'hui librement accessibles. L'un des plus utilisés est dolly 15k¹², qui contient 15 000 exemples d'instructions créées par Databricks, une entreprise spécialisée dans le développement d'une plateforme cloud d'analyse et de traitement de données pour faciliter les solutions d'analyse et d'apprentissage automatique à grande échelle. Il est probable que les entreprises à l'origine des principaux giga-modèles disposent de corpus propriétaires encore plus importants, réservés à leur usage exclusif.

Cette étape d'affinage supplémentaire améliore les modèles de deux manières. Comme on pouvait s'y attendre, les modèles donnent de meilleures réponses lorsqu'ils sont confrontés à des amorces présentes dans le corpus utilisé pour l'affinage. Étant donné la taille de ces bases de données, il est probable qu'elles renferment quantité d'amorces que les utilisateurs soumettront, ce qui renforce la capacité du modèle à apporter une réponse convaincante, surtout dans un contexte *zero-shot* dans lequel le modèle n'a accès à aucun exemple de la tâche à effectuer.

De manière plus surprenante, plusieurs études ont montré que cette étape améliorerait également les réponses du modèle face à de nouvelles amorces qui n'avaient pas été vues pendant l'affinage. Il semble, en effet, que le modèle acquière, au cours de l'affinage, des compétences *génériques* et qu'il apprenne à répondre à des questions plutôt que de *juste* générer du texte.

Prendre en compte les jugements humains

Malgré toutes leurs qualités, les modèles de langues souffrent d'un écueil majeur : bien que ceux-ci apprennent à générer du texte syntaxiquement et sémantiquement correct en cherchant à prédire les mots masqués lors de l'apprentissage, ils ne reçoivent aucune information pour apprendre à répondre à une question ou à une amorce en particulier.

Il faut donc *aligner* les textes générés par des modèles de langue avec l'intention exprimée dans une amorce, exactement comme on a aligné les représentations des textes et des images. Pour cela, il est nécessaire de disposer d'une information supplémentaire sur les réponses générées par un modèle. Ces avis sont généralement recueillis sous forme binaire (la réponse du système est-elle adéquate ou non ?) soit directement auprès de l'utilisateur soit auprès d'un annotateur recruté spécifiquement pour évaluer les réponses.

La prise en compte des jugements des utilisateurs s'inscrit dans un domaine distinct de l'apprentissage statistique, connu sous le nom d'*apprentissage par renforcement*. C'est typiquement le type d'apprentissage utilisé dans les systèmes de conduite autonome, AlphaGo et de nombreux jeux, mais aussi dans les systèmes de placement de publicités qui génèrent la majorité des revenus des principaux acteurs du web.

Ce type d'apprentissage diffère fondamentalement de l'apprentissage supervisé, sur lequel s'appuient toutes les méthodes que nous avons vues jusqu'à présent. Dans l'apprentissage supervisé, l'algorithme a accès à la « bonne » solution pour chaque exemple du corpus d'apprentissage, ce qui lui permet d'adapter ses poids pour reproduire les sorties qu'un expert aurait produites. En revanche, dans l'apprentissage par renforcement, l'algorithme ne dispose que d'informations limitées : il sait seulement si la décision qu'il a prise est bonne ou non (selon le jugement émis), sans savoir quelle aurait été la bonne réponse (celle donnée par être humain).

De nombreuses méthodes ont été proposées pour recueillir les jugements des utilisateurs sur les sorties générées et les intégrer dans le processus d'apprentissage. L'approche la plus couramment utilisée aujourd'hui (essentiellement parce qu'elle est utilisée par OpenAI) est appelée « renforcement humain ». Elle permet, lors d'une phase suivant l'apprentissage habituel des modèles de langue, de mettre à jour l'ensemble des poids d'un modèle en fonction des avis reçus, de manière à ce que ses prédictions soient de plus en plus alignées avec les jugements humains à mesure qu'il reçoit des évaluations de ses sorties. À la fin, les sorties générées devraient correspondre exactement aux préférences des utilisateurs.

Cette étape d'alignement permet également de garantir la qualité des textes générés par les giga-modèles. Même si ces textes sont depuis longtemps crédibles (sans erreurs de syntaxe majeures ni incohérences sémantiques), ils ne sont pas nécessairement vrais (ils peuvent contenir des informations incorrectes), fiables (le modèle produit des réponses similaires pour des entrées similaires) ou éthiques (ils peuvent contenir des propos inappropriés, par exemple racistes). Lors de l'étape de renforcement humain, un texte généré ne respectant pas ces trois critères recevra un jugement négatif qui permettra au système d'apprendre progressivement à ne plus générer ce type de contenu.

Le renforcement humain est donc une étape nécessaire pour garantir la qualité des textes générés, éviter certains des problèmes rencontrés par les premiers chatbots, et assurer que les réponses correspondent bien à l'amorce. Cependant, sa mise en œuvre nécessite la collecte de jugements humains, ce qui nécessite soit d'avoir le budget pour recruter des annotateurs, soit de disposer d'une base d'utilisateurs suffisamment importante pour recueillir leurs retours. Dans les deux cas, c'est un frein sérieux au développement des giga-modèles.

Et ChatGPT dans tout ça ?

Nous avons maintenant décrit en détail les trois briques sur lesquelles ChatGPT est bâti. La première brique est constituée de la paire transformeurs (le réseau de neurones au cœur de tous les modèles que nous avons vus) et apprentissage auto-supervisé (qui permet de construire des modèles de langue). La deuxième est l'augmentation de la taille des corpus et du nombre de paramètres qui a fait émerger des capacités étonnantes dans ces modèles. La troisième est l'affinage de ces giga-modèles sur des corpus d'instructions, additionnée de la prise en compte des jugements humains.

Le succès de ChatGPT et ses spectaculaires capacités ne sont, une fois de plus, pas dus à une nouvelle découverte ou à une avancée technique majeure. Tous les éléments « techniques » sur lesquels repose le chatbot étaient déjà au cœur d'InstructGPT ! L'innovation est plutôt à chercher du côté des données : ChatGPT n'est qu'une version affinée d'InstructGPT, conçue spécifiquement pour dialoguer avec des humains.

Et c'est là l'idée géniale d'OpenAI (en plus d'avoir rendu accessible son modèle au grand public) : la société a créé une interface à la fois simple et intuitive pour utiliser ces puissants outils que sont les giga-modèles. Plus besoin d'écrire des programmes pour envoyer les amorces aux serveurs d'OpenAI et analyser leurs réponses. Utiliser une interface similaire à celle de n'importe quel système de messagerie instantanée et poser des questions comme si l'on dialoguait avec un ami est tout ce qu'il faut faire.

Pour doter un giga-modèle de la capacité de dialoguer, il n'est pas nécessaire de créer un nouveau modèle : il suffit de disposer d'exemples de dialogues ou d'avis humains pour utiliser les méthodes d'affinage. Aucune *intelligence* n'a été utilisée pour permettre à GPT-3 de dialoguer. OpenAI a simplement engagé un

grand nombre d’annotateurs qui ont consacré des heures à rédiger les réponses attendues et à évaluer les réactions du modèle.

Comme nous en avons déjà discuté au chapitre 2 (voir [ici](#)), il y a de nombreux problèmes liés à l’annotation de données par des travailleurs du clic. Les choses ne sont pas différentes pour ChatGPT : une enquête retentissante du *Time*, publiée en janvier 2023¹³, a révélé que ce travail d’annotation a été principalement effectué par des annotatrices et des annotateurs kényans très peu rémunérés (en moyenne 1,42 dollar de l’heure) et exposés à des contenus particulièrement violents et toxiques. En octobre 2023, d’anciens annotateurs ayant travaillé sur les données utilisées par ChatGPT ont déposé une requête devant le Parlement kényan pour dénoncer leurs conditions de travail¹⁴.

Même si peu d’informations sont disponibles sur la quantité d’annotations recueillies, on sait que pour le développement de ChatGPT, OpenAI a passé un contrat de 200 000 dollars avec SAMA, une entreprise spécialisée dans l’annotation de contenu. En supposant que la moitié de ce montant soit consacrée au salaire des annotateurs, et en prenant comme base la rémunération de 2 dollars de l’heure mentionnée dans l’article du *Time*, on peut estimer (de manière grossière) qu’il y a eu au moins 50 000 heures (presque 6 ans !) consacrées à l’annotation. Ce travail en amont (pour filtrer les données d’apprentissage et supprimer tout contenu potentiellement toxique) et en aval (pour évaluer les réponses du modèle), a permis à la fois à ChatGPT d’apprendre à dialoguer mais aussi à ne pas tenir de propos inappropriés.

Malgré les prouesses des transformeurs et des giga-modèles, la capacité de l’ordinateur à dialoguer est donc encore fortement liée à au travail des petites mains de l’IA. Comme nous allons maintenant le voir, ce n’est pas là la seule limite de ce modèle.

CHATGPT ET LES GIGA-MODÈLES, UNE RÉVOLUTION OU UNE ÉVOLUTION ?

Les giga-modèles représentent un véritable changement de paradigme. Ils permettent aux ordinateurs d'accomplir des tâches pour lesquelles ils n'ont pas été programmés et que l'on pensait jusque très récemment réservées aux humains. Quelles sont ces tâches ? Quel impact aura l'usage croissant de ces modèles sur notre vie quotidienne ? Peut-on affirmer que les machines sont devenues intelligentes ?

Une seule limite : l'imagination

Construits autour de deux idées très simples (prédire le mot suivant dans une phrase et aligner les textes générés avec les jugements humains), les giga-modèles semblent pourtant offrir des possibilités infinies. La sortie de ChatGPT en novembre 2022 a permis à un large public (le service comptait déjà un million d'utilisatrices et d'utilisateurs 5 jours après son lancement, un nombre qui dépasse aujourd'hui les 200 millions) de découvrir et de tester les possibilités offertes par l'IA générative.

L'impact sur la société a été considérable. Qu'il s'agisse de la presse grand public (tout de même 20 000 articles entre novembre 2022 et juin 2023 ¹) ou de tout un chacun autour de la machine à café ou sur les réseaux sociaux, tout le monde s'est émerveillé devant les possibilités offertes par ChatGPT. On a vanté la qualité de ses réponses tout autant que décrié ses erreurs.

Cet engouement a fait une publicité monstre à OpenAI et a braqué les feux des projecteurs sur cette start-up jusque-là connue uniquement des spécialistes de l'IA : en septembre 2023, la valorisation boursière d'OpenAI était estimée à 90 milliards de dollars², un montant qui a triplé depuis le lancement de ChatGPT. Aujourd'hui, c'est l'une des start-up les plus valorisées, derrière SpaceX ou ByteDance (le propriétaire de TikTok).

Il existe de nombreux sites et même des livres expliquant en détail toutes les utilisations possibles de cet outil. Plutôt que l'exhaustivité, nous allons ici chercher à décrire quelques exemples représentatifs. Un premier type d'utilisation, qui est très vite apparu, est l'aide à l'écriture : à partir d'un premier jet, ChatGPT corrigera les fautes, améliorera la lisibilité, réduira (ou, suivant ce qu'on lui demande, allongera) le texte, ajoutera les formules de politesse ou reformulera un texte pour le rendre plus formel ou écrire selon un style particulier. Il peut également rédiger directement des e-mails ou des courriers à partir d'une simple description de l'intention du message (*peux-tu écrire un mail formel pour préciser que la date de rendu du projet est dans 8 jours et qu'il n'y aura pas de délais ?*). Si le gain de temps est déjà appréciable dans sa langue maternelle, ChatGPT, parfaitement multilingue, fournit également une aide à la rédaction dans une langue étrangère (il suffit d'ajouter « en anglais » ou en « en grec » à la demande pour changer la langue de la réponse), ce qui ouvre de nouvelles possibilités de communication. Les giga-modèles sont des correcteurs si performants que plusieurs journaux ont déjà annoncé étudier la possibilité de les utiliser pour épauler les secrétaires de rédaction dans leur travail de correction³.

ChatGPT a également très vite servi à retrouver de l'information, remettant en cause l'interface traditionnelle des moteurs de recherche. Plutôt que de renvoyer une liste de documents jugés pertinents pour une requête, laissant l'utilisateur extraire l'information recherchée et la reformuler, ChatGPT peut répondre directement à des questions précises à partir des connaissances

qu'il a extraites lors de son apprentissage (*Quand est né Molière ? Quel est le plus haut bâtiment construit dans l'Antiquité ?*) mais également écrire, sur un sujet donné, une véritable synthèse dont il pourra adapter la longueur, la forme ou le degré de précision selon les indications de l'utilisateur (*Peux-tu me décrire les grandes étapes dans le développement des chatbots en 500 mots ?*).

ChatGPT peut aussi résumer des documents de plusieurs pages en quelques phrases ou paragraphes (il suffit de lui demander la taille voulue), ou croiser des informations, par exemple, pour identifier les fournisseurs avec la plus petite empreinte carbone⁴ ou extraire des données d'un ensemble de documents, voire écrire des mangas⁵ (aussi bien le scénario que les images).

ChatGPT est également un aide-cuisinier pour créer des recettes originales à partir des ingrédients disponibles dans le réfrigérateur⁶ (*ChatGPT, que mange-t-on ce soir ?*). Il peut résoudre des problèmes de mathématiques et de physique, rédiger des devoirs et des dissertations d'étudiants⁷, ainsi que des rapports médicaux, bien que des erreurs puissent survenir dans ces derniers cas. Le modèle a réussi le test de codage informatique que fait passer Alphabet à tous ses futurs employés ; il possède donc les compétences d'un ingénieur de niveau 3 payé avec un salaire annuel de 183 000 dollars⁸.

Il faut dire que ChatGPT est parfaitement capable de générer un programme à partir d'une description en langage naturelle (*Est-ce que tu pourrais écrire un programme python qui cherche le mot le plus fréquent dans les fichiers Word contenus dans un répertoire ?*). La possibilité pour un ordinateur de générer du code à partir d'une description n'est pas surprenante : cela fait des années qu'il existe des forums sur lesquels les informaticiens s'échangent des informations, exposent leurs problèmes et discutent des solutions. Par exemple, StackOverflow, l'un des forums les plus utilisés aujourd'hui, contient (en octobre 2023) plus de 24 millions de questions et 35 millions de réponses couvrant une très grande

partie des problèmes que l'on rencontre lors du développement d'un programme informatique. Toutes ces ressources ont probablement été intégrées dans l'apprentissage de ChatGPT. Il suffit très souvent d'écrire la question posée à ChatGPT dans un moteur de recherche *classique* pour trouver des pages contenant tous les éléments nécessaires pour réaliser le programme demandé. La capacité de ChatGPT à synthétiser ces réponses reste toutefois remarquable.

L'usage de ChatGPT semble infini. Mais il est important de noter que juger les capacités et l'utilité d'un programme, en particulier dans les domaines du TAL et de l'IA, uniquement à partir d'anecdotes et d'une poignée de réussites, est problématique : personne n'a envie d'utiliser, les yeux fermés, un système de traduction automatique juste parce qu'il a traduit correctement deux ou trois phrases.

Et effectivement, des articles plus critiques ont déjà commencé à fleurir, même s'ils ont reçu moins d'échos que les louanges. Par exemple, en octobre 2023, le journal *Le Monde* publiait un article⁹ qui résumait plusieurs travaux scientifiques ayant analysé de manière plus systématique les performances de ChatGPT dans divers domaines et présente une conclusion sans appel : « Ces résultats font retomber un peu le soufflé. »

Avant même que les capacités de ChatGPT ne soient critiquées, les discours enthousiastes sur l'IA générative ont très vite été accompagnés de discours inquiets sur les risques liés à son utilisation. À mesure qu'on prenait conscience de la toute-puissance des giga-modèles, des craintes se sont exprimées sur le possible remplacement des humains par une machine (une angoisse présente dès les débuts de l'IA, dans les années 1950), et surtout sur les menaces pesant sur l'emploi.

Après avoir expliqué le fonctionnement de ce système et évoqué ses utilisations possibles, il est maintenant temps d'aborder ses limites. Celles-ci sont de deux types : celles que je qualifierais de

temporaires, liées à ChatGPT et au fait qu'il soit le fruit d'une grande entreprise, et les limites *fondamentales*, résultant du principe même de l'apprentissage auto-supervisé d'une langue.

Les limites propres à ChatGPT

Deux critiques récurrentes faites à ChatGPT (mais qui sont valables pour tous les giga-modèles) portent sur le fait qu'il soit ignorant des événements récents et qu'il fasse des erreurs. Il est en effet clair que ChatGPT ne sait rien de ce qui s'est passé après janvier 2022, date à laquelle son corpus d'entraînement a été constitué. Cette information est clairement spécifiée sur la page du chatbot et rappelée à chaque fois qu'une question sur un événement postérieur à cette date est posée. Impossible de lui demander quels sont les derniers prix Nobel ou de l'interroger sur le dernier film d'un réalisateur.

Cette critique découle directement de la nature des giga-modèles : contrairement aux moteurs de recherche qui s'appuient sur les documents présents sur l'Internet pour répondre à une requête et qui peuvent être mis à jour quasiment en temps réel, ChatGPT s'appuie sur les connaissances qu'il a extraites de son corpus d'apprentissage. Il n'est donc pas surprenant qu'il ne puisse pas intégrer des informations postérieures à cette opération. Ce n'est tout simplement pas le bon outil pour accéder à des informations récentes.

La critique sur les erreurs faites par ChatGPT est également directement liée à la *nature statistique* de ce modèle : comme tous les systèmes reposant sur l'apprentissage statistique, il n'y a aucune garantie que ChatGPT donne la bonne réponse dans tous les cas. Comme expliqué au chapitre 3 (voir [ici](#)), la théorie de l'apprentissage statistique n'apporte qu'une seule garantie : changer un seul paramètre de ChatGPT dégraderait les performances du chatbot qui donnerait, en moyenne, de moins bonnes réponses que celles du système actuel. Il est donc illusoire

d'espérer que ChatGPT ne se trompe jamais, tout comme il est illusoire d'espérer qu'un filtre de spam ne fasse jamais d'erreur ou qu'un système de traduction automatique traduise parfaitement n'importe quelle phrase.

C'est un changement de paradigme majeur en informatique : l'un des enjeux centraux de l'informatique a toujours été de garantir que les algorithmes mis au point par les informaticiens fonctionnent correctement dans tous les cas. L'idée d'un programme qui puisse donner de mauvaises réponses sort complètement de ce cadre et apparaît tout à fait étrange. Mais c'est la nature même de l'apprentissage statistique !

En plus de ces deux critiques directement liées à la nature du modèle, deux autres reproches sont fréquemment adressés au chatbot et aux principaux giga-modèles : leur coût et les inquiétudes qu'ils soulèvent vis-à-vis du transfert de données personnelles. Ces deux points méritent d'être discutés en détail.

Confidentialité des données et souveraineté numérique

Une des premières limites de ChatGPT historiquement pointée concerne les questions de souveraineté des données et de propriété intellectuelle : le système fonctionne sous forme de *service*. Les amorces et les questions soumises par un utilisateur à partir de son navigateur ou de l'application mobile sont envoyées sur un serveur d'OpenAI (dont 49 % du capital est maintenant détenu par Microsoft, après que le géant du logiciel a investi plus de 10 milliards de dollars dans la startup et fourni une grande partie de l'infrastructure informatique de celle-ci), qui renvoie ensuite sa réponse. En pratique, cela signifie que toutes les requêtes soumises à ChatGPT peuvent être (et sont très certainement) analysées voire stockées par l'entreprise et peuvent (et très certainement vont) servir à alimenter les corpus utilisés pour entraîner les prochains modèles de langues. C'est particulièrement problématique lorsqu'on utilise un giga-modèle

pour corriger, traduire ou simplifier un mail, un contrat en cours de négociation ou une demande de brevet qui contiennent des données potentiellement confidentielles. De nombreuses entreprises (Samsung, Goldman Sachs, etc.) ont décidé d'interdire à leurs employés d'utiliser tout service d'IA de manière préventive ou à la suite de fuites avérées de données confidentielles¹⁰.

À l'heure où ces lignes sont écrites, OpenAI offre la possibilité à ses utilisateurs de ne pas inclure leurs requêtes dans ses bases d'entraînements (ce qui est donc le cas par défaut). Si c'est un pas dans la bonne direction, cela n'empêche pas l'entreprise d'analyser toutes ces requêtes, et il ne faut pas oublier que, depuis les révélations de Edward Snowden en 2013, nous savons que les services de sécurité américains (typiquement la NSA) disposent d'un accès direct aux données hébergées par les géants américains des nouvelles technologies, et notamment toutes les grandes entreprises (Alphabet, Meta, Microsoft, etc.) qui développent des giga-modèles.

Les problèmes de confidentialité se posent même quand ChatGPT est utilisé uniquement dans un cadre personnel pour choisir son repas ou écrire le discours d'un mariage. Si les informations communiquées à OpenAI/Microsoft sont moins sensibles que des secrets industriels, il ne faut pas perdre de vue que Microsoft, comme toutes les entreprises du web, tire une part non négligeable de ses revenus de sa capacité à placer des publicités ciblées adaptées au profil des utilisateurs de ses services. À l'âge de ce que Shoshana Zuboff appelle le « capitalisme de surveillance »¹¹, la capacité des entreprises du web à constituer des profils très détaillés de leurs utilisateurs est une priorité majeure. En exigeant la création d'un compte personnel associé à un numéro de téléphone, Microsoft/OpenAI moissonnent des informations personnelles qu'elles sauront valoriser, soit directement en ciblant mieux la publicité, soit en les revendant à d'autres entreprises.

La question des données et de leur exploitation se pose également par rapport aux corpus d'apprentissage des giga-modèles : comme nous l'avons vu, ces systèmes sont appris à partir de vastes corpus constitués en grande partie de pages web. Or il est plus que probable que certaines renferment des données personnelles ou soient protégées par des droits d'auteur, et que de manière involontaire, ChatGPT divulgue ces informations. Des chercheurs ont ainsi réussi à montrer que les corpus utilisés pour entraîner ChatGPT (mais aussi plusieurs giga-modèles d'Alphabet et de Meta) contenaient des livres téléchargés à partir des principaux annuaires d'ouvrages piratés et que leur contenu apparaissait dans ses réponses.

Suite à ces révélations, l'Authors Guild, l'une des principales organisations d'auteurs américains, a déposé plainte contre OpenAI en septembre 2023¹². Plusieurs célébrités, dont George R. R. Martin et Jodi Picoult, se sont associées à cette plainte. Les reproches faits à OpenAI sont nombreux, mais portent principalement sur « les violations flagrantes et préjudiciables des droits d'auteur enregistrés des plaignants sur les œuvres écrites de fiction » et l'utilisation par OpenAI de leurs ouvrages « sans autorisation ni contrepartie ». Des plaintes similaires ont été déposées par les artistes qui voient leurs œuvres reproduites par DALL·E. Le risque d'un procès est tel que Microsoft a annoncé, en septembre 2023, couvrir ses clients dont les contenus générés par des outils d'IA seraient jugés être des contrefaçons¹³.

Ce sont pour les mêmes raisons que de nombreux médias dans le monde (en France, Radio France, TF1, France Média Monde et bien d'autres) ont bloqué l'accès à leurs sites aux principales entreprises développant des giga-modèles pour éviter que leurs contenus soient exploités sans contrepartie¹⁴. Les supports de presse sont en effet une source de données importante tant par la qualité des textes (en général tous écrits par des professionnels et

relus par des éditeurs expérimentés) que par les informations qu'ils contiennent (sujet d'actualité, analyses détaillées, etc.).

Les données sont aujourd'hui l'enjeu de nombreuses convoitises, d'un côté les concepteurs des giga-modèles cherchent à augmenter la taille de leurs corpus d'apprentissage, et de l'autre, les journalistes, et plus généralement les ayants-droits, veulent éviter le pillage de leurs données et obtenir des compensations financières. Fin décembre 2023, peu de temps après qu'Axel Springer, l'un des principaux groupes de presse allemand, a annoncé avoir signé un accord donnant accès à OpenAI à l'ensemble de ses contenus contre une rétribution de plusieurs dizaines de millions de dollars, le *New York Times* a porté plainte pour violation des droits d'auteur contre OpenAI, marquant un nouveau tournant dans la guerre opposant les créateurs de contenus aux développeurs de ChatGPT¹⁵.

L'enjeu dépasse la question des droits d'auteur. Pour preuve de l'importance du sujet, Emmanuel Macron lui-même a souligné fin juin 2023 les « biais natifs anglo-saxons » dont souffrent ChatGPT et la plupart des modèles¹⁶ : ceux-ci ont, par exemple, tendance à présenter le 4 juillet comme un jour de fête, quelle que soit la langue dans laquelle il s'exprime, un comportement qu'il n'a pas avec le 14 juillet. Il faut dire que, selon Jean-Noël Barrot, le ministre délégué chargé de la transition numérique et des télécommunications, le français ne représente que 0,2 % des données d'apprentissage du chatbot ! Pour défendre le « soft power » français, selon les mots du Président, la direction interministérielle du numérique a été chargée de créer une grande base de données appelée « Villers-Cotterêts » (le lieu où fut signée l'ordonnance faisant du français la langue officielle de notre État), qui regroupera et rendra accessible aux industriels de l'IA une vaste quantité de textes en français, principalement issus de la BNF, de l'INA et des archives nationales. À l'heure où ces lignes sont écrites, les discussions sont en cours pour mettre en œuvre ce

projet et trouver des solutions à la question épineuse de la rétribution des auteurs.

Enfin, il est essentiel de prendre conscience que plus ChatGPT sera utilisé, que ce soit par des individus ou des entreprises, plus ceux-ci deviendront dépendants de ce service, et donc de Microsoft/OpenAI. Le jour où cette entreprise décidera d'augmenter ses prix, voire de réserver l'utilisation de son service à ses seuls clients payants, cet assujettissement risque de poser de graves problèmes. Il est également tout à fait envisageable que Microsoft/OpenAI décide d'arrêter ce service, que ce soit pour des raisons de coût ou par respect de décisions politiques américaines. N'oublions pas qu'en 2019, l'administration du président Trump a interdit aux entreprises américaines de commercer avec le chinois Huawei, l'un des principaux fabricants mondiaux d'équipements de réseaux et de téléphonie mobile, entraînant un bouleversement sans précédent dans le monde des nouvelles technologies : du jour au lendemain (ou presque), plus aucun service d'Alphabet (ni Google Maps ni Play Store, le principal moyen d'installer des applications sur un téléphone Android) ne pouvait être utilisé sur un téléphone de la marque chinoise¹⁷.

Les coûts cachés des giga-modèles

Une deuxième critique récurrente des giga-modèles concerne leurs frais de fonctionnement. Utiliser ChatGPT est aussi simple que d'écrire un SMS (et c'est là tout son intérêt). Pourtant, la réponse à ce message nécessite un nombre faramineux de calculs, même si, comme souvent en informatique, le coût associé est complètement invisible aux yeux des utilisateurs. Comme je l'ai expliqué au chapitre 4 (voir [ici](#)), les capacités des giga-modèles, et en particulier de ChatGPT, dépendent fortement de leur taille. Ce sont les milliards de paramètres qui les constituent et les milliards de mots du corpus d'apprentissage qui ont fait émerger leurs capacités. Apprendre et utiliser des réseaux de neurones de cette

taille sur une telle quantité de données nécessite une infrastructure informatique conséquente, notamment de (très) nombreux GPUs particulièrement chers et énergivores.

Il est difficile de quantifier exactement les coûts d'un giga-modèle et, notamment, leur empreinte écologique. Les infrastructures utilisées sont considérées comme des secrets industriels, et les différents acteurs du marché communiquent peu sur ces questions. Une étude¹⁸ associant des universitaires à des chercheurs d'Alphabet a conclu que le coût, notamment énergétique, des giga-modèles était trop élevé et les problèmes soulevés trop nombreux pour justifier la course dans laquelle Alphabet s'était lancé. Le rapport a entraîné le licenciement, très médiatisé, de Timnit Gebru et Margaret Mitchell, deux chercheuses de l'équipe d'éthique d'Alphabet, peu enclin à voir sa stratégie ainsi critiquée¹⁹.

Chercheurs et analystes ont tenté ces dernières années d'avoir une idée un peu plus précise sur le problème. L'entraînement d'un seul modèle BERT, l'ancêtre de tous les giga-modèles, a émis 284 tonnes équivalent CO₂, soit les émissions de 57 humains *moyens* pendant un an ou d'un avion volant de New York à Los Angeles. Il ne faut pas oublier que la mise au point d'un modèle de langue nécessite généralement de très nombreux entraînements pour mettre au point l'architecture optimale et régler les différents hyper-paramètres contrôlant l'apprentissage. Les chiffres sont, comme on pouvait s'y attendre, encore plus impressionnants pour l'apprentissage d'un giga-modèle. On estime ainsi que l'entraînement de GPT-3 a nécessité 1,3 GWh d'électricité, soit l'équivalent de la consommation annuelle de 260 foyers français de quatre personnes, générant 502 tonnes équivalent carbone²⁰.

Si ces chiffres sont à prendre avec des pincettes, car ils dépendent de nombreux paramètres qui ne sont pas toujours connus (matériel utilisé lors de l'apprentissage, localisation des data

centers, type d'énergie utilisée pour les alimenter, etc.), ils montrent que l'apprentissage des giga-modèles a un coût écologique important. Et encore, nous n'avons abordé qu'un seul aspect : il ne faut pas oublier que de nombreux métaux rares (lithium, germanium, etc.) entrent dans la fabrication de leurs composants, et que l'extraction de ces matériaux soulève également de nombreux problèmes écologiques.

Le coût de la mise au point d'un giga-modèle (l'entraînement de GPT-4 a duré entre 90 et 100 jours et a mobilisé 25 000 GPUs) expliquent pourquoi ils ne sont pas ré-entraînés plus souvent (et donc pourquoi ChatGPT continue à ne pas connaître les événements récents). Pour autant, les montants en jeu sont négligeables vis-à-vis des frais de fonctionnement. Pour répondre aux millions de requêtes quotidiennes, ChatGPT s'appuie sur 29 000 CPUs, et OpenAI dépense chaque jour 700 000 dollars pour faire tourner cette infrastructure, bien en deçà des bénéfices qu'il tire de ce service²¹. Ces chiffres paraissent gigantesques, mais ils sont bien inférieurs aux sollicitations des moteurs de recherche traditionnels. Google répond à 320 000 requêtes par seconde. Si OpenAI devait traiter autant de demandes, il lui faudrait plus de 4 milliards de GPUs. L'achat de ces serveurs et la création de l'infrastructure pour les utiliser nécessiteraient un investissement qui dépasserait les 100 milliards de dollars, sans parler des frais d'exploitation.

Les coûts liés à l'inférence des giga-modèles expliquent très certainement pourquoi si peu d'acteurs sont aujourd'hui prêts à lancer un concurrent à ChatGPT, même si de plus en plus de modèles alternatifs ont été développés. Ils jettent également des doutes sur la capacité d'OpenAI à garder ouvert l'accès à ChatGPT, voire sur le risque d'un nouvel hiver de l'IA. Nous reviendrons sur cette question.

Le manque de transparence

Le manque de transparence sur les données utilisées lors de l'apprentissage est une autre des critiques récurrentes adressées à ChatGPT.

Comme nous en avons déjà longuement parlé, les giga-modèles nécessitent un apprentissage sur d'immenses corpus de données (on parle quand même, pour GPT-4, d'un corpus contenant un trillion de mots, soit 3,5 millions de fois plus que la totalité des mots composant l'ensemble des tomes de *Game of Thrones*). La taille de ce corpus est souvent mise en avant pour justifier la capacité des giga-modèles à s'exprimer correctement, et surtout pour démontrer l'étendue de leurs connaissances : les giga-modèles seraient parvenus, d'une certaine manière, à stocker et à manipuler une version compressée de l'ensemble des connaissances humaines (ils ont *tout lu* !).

Cependant, plus que la quantité de données, la qualité de ces données joue un rôle essentiel dans la capacité des modèles à fournir des réponses de qualité. Il est crucial de ne pas oublier que les corpus d'apprentissage des giga-modèles sont filtrés et que leur contenu est soigneusement choisi en fonction de sa qualité. Finalement, un grand nombre de documents (sans que l'on puisse savoir lesquels) ne sont jamais considérés lors de l'entraînement des modèles : ce n'est pas tant la totalité de l'Internet qui a servi à entraîner les modèles qu'un sous-ensemble ne reflétant que les opinions et les connaissances d'une toute petite partie de la population. Et encore ! Il est tout à fait possible qu'une grande partie des informations extraites des pages sélectionnées ne soit pas utilisée car elle n'est pas alignée avec les jugements des utilisateurs.

Aujourd'hui, très peu d'informations sont disponibles sur les données à partir desquelles les giga-modèles sont appris (aussi bien le modèle de langue que les jugements des utilisateurs), ce qui complique l'identification des biais et des limites de ces modèles. Ce manque de transparence permet de protéger des

secrets industriels : l'exemple de BLOOM, dont nous avons parlé précédemment (voir [ici](#)), montre l'importance du filtrage sur la qualité des textes produits. Mais c'est aussi (et surtout) un moyen d'éviter que les créateurs des textes utilisés lors de l'apprentissage puissent être identifiés et exercent leurs droits d'auteur pour obtenir une rétribution.

Le filtrage de certaines sources d'information n'est pas nouveau (les moteurs de recherche, tels que Google, filtrent depuis des années leurs réponses pour éviter la diffusion de contenu violent ou raciste ou en réponse à des décisions judiciaires). Mais, différence fondamentale, celui-ci est dissimulé dans les giga-modèles. Ainsi, lorsqu'un giga-modèle répond « je ne peux pas vous fournir d'aide » ou « je ne sais pas répondre à cette question », il sous-entend en réalité quelque chose qui pourrait ressembler à « mes concepteurs ne m'ont pas fourni ces informations » ou « les jugements de mes concepteurs m'interdisent de répondre ». Le choix délibéré des concepteurs de ChatGPT et de nombreux autres giga-modèles (parce que c'est bel et bien un choix de conception assumé) est de privilégier le premier type de réponse plutôt que le second. Ce choix dévoile leur volonté de minimiser, voire de cacher, les interventions humaines sur les textes produits par des IA génératives. Cela confie à ces textes une plus grande valeur : la réponse a été produite par une IA mue uniquement par la logique froide et implacable des statistiques, et non par un humain plein de biais et de préjugés.

Ce ne sont pas les seules interventions humaines dont l'importance pour les giga-modèles est minimisée : les textes utilisés pour leur apprentissage ont tous été rédigés par des personnes. De nombreux travaux et retours d'expérience montrent que l'inclusion de contenu généré automatiquement dans les données d'apprentissage d'un giga-modèle dégrade considérablement la qualité de ses productions. La création de tous les corpus d'apprentissage actuels commence par la sélection

de sources dont on est certain que le contenu est humain. De nombreux professionnels tels que journalistes, écrivains, scénaristes, juristes, et bien d'autres, ont l'impression que leur production est utilisée sans rétribution pour alimenter les IA. Nous avons déjà mentionné les procès en cours contre OpenAI, et il ne fait aucun doute que les décisions rendues auront un impact sur le développement des giga-modèles. La production des textes utilisés pour l'entraînement des modèles n'est pas la seule phase du développement des giga-modèles qui exploite le travail humain. Bien que ces modèles soient souvent présentés comme apprenant de manière complètement non supervisée, il ne faut pas négliger, comme mentionné précédemment, l'énorme travail d'annotation nécessaire pour filtrer le corpus d'apprentissage et fournir des jugements sur les sorties du modèle.

En fin de compte, si les giga-modèles fonctionnent aussi bien, c'est en grande partie grâce au travail humain, souvent invisibilisé. Une fois de plus, les efforts déployés pour dissimuler toutes ces interventions humaines reflètent l'intention délibérée des concepteurs de minimiser le rôle des humains dans le progrès des giga-modèles et plus généralement de l'IA. Il est sans aucun doute plus flatteur et vendeur (surtout vis-à-vis des investisseurs) de justifier les progrès d'un algorithme par des avancées scientifiques et la maîtrise d'une technologie complexe plutôt que par un investissement conséquent dans la collecte et l'annotation de données, surtout lorsque cela repose essentiellement sur l'exploitation de travailleurs.

Les limites de l'apprentissage auto-supervisé

Si toutes les craintes et les limites que nous avons évoquées jusqu'à présent sont légitimes et concernent des problèmes importants, elles ne sont à mes yeux que temporaires. De nombreuses équipes de recherche travaillent aujourd'hui sur des méthodes visant à réduire le nombre de paramètres d'un modèle

(notamment à l'aide de techniques de *distillation* qui consistent à simplifier un grand modèle en réduisant le nombre de paramètres ou de *quantification*, une méthode de compression qui permet de réduire la taille des modèles) ou à accélérer l'apprentissage, par exemple en permettant la mise à jour du modèle dès que de nouvelles données sont disponibles plutôt que de recommencer l'apprentissage à zéro (techniques dites *d'apprentissage continu*). Il existe même désormais des méthodes d'affinage comme LoRA qui réduisent significativement la complexité de cette opération en identifiant les paramètres les plus importants du modèle et en *concentrant* les mises à jour sur eux. L'affinage peut être jusqu'à 10 000 fois plus rapide !

Les premiers résultats sont particulièrement encourageants. LLaMA, un modèle développé par Meta accessible sous une licence particulièrement permissive, peut être utilisé pour générer du texte sur un ordinateur portable un peu puissant mais tout à fait accessible au grand public (typiquement un MacBook Pro). Ses performances sont jugées comparables à celles de GPT-3. Vu le nombre et la diversité des chercheurs travaillant sur ces questions, et le nombre de concurrents à ChatGPT qui se développent²², aussi bien dans les grandes entreprises que dans les laboratoires de recherche ou dans des projets open source, il est fort probable qu'à moyen terme (et je serais presque prêt à parier que cela arrivera plutôt tôt que tard), il soit possible de déployer des modèles de langues adaptés aux besoins d'une entreprise (voire d'une personne) *en local*, et non sur des serveurs contrôlés par une entreprise tierce. Ces prévisions semblent se concrétiser : en septembre 2023, Mistral AI, une start-up française dont nous avons déjà parlé dans l'introduction, a publié son premier giga-modèle, Mistral 7B, comprenant 7 milliards de paramètres sous une licence libre, permettant à quiconque de le télécharger, de l'utiliser et de l'affiner tout en garantissant la confidentialité des données. Ce modèle intègre plusieurs innovations qui lui permettent d'obtenir d'excellentes

performances dans de nombreux tests officiels. Les prochains progrès des giga-modèles ne viendront peut-être pas des GAFAM. Si les giga-modèles personnalisés et la multiplication des acteurs peuvent apporter une solution à tous les problèmes que nous venons d'évoquer, ils continuent de heurter certaines limites qui découlent directement de la manière dont ces systèmes acquièrent leur capacité à manipuler la langue au travers de l'apprentissage auto-supervisé. Ces limites sont donc nettement plus fondamentales.

Forme et sens à l'âge des giga-modèles

L'un de ces défauts constitutifs concerne l'incapacité des giga-modèles à comprendre le sens. Fondamentalement, les modèles de langue sont entraînés à partir d'un critère d'apprentissage consistant à retrouver des mots masqués. Cette tâche leur permet de découvrir et d'exploiter les relations entre les mots qu'ils manipulent, et ainsi de générer un texte syntaxiquement et sémantiquement correct.

Mais pour réussir à prédire des mots masqués, un modèle n'a besoin que de savoir manipuler des symboles (les suites de caractères qui composent les mots). Si le choix des paramètres durant l'apprentissage permet aux modèles de langue d'apprendre que certains mots partagent des propriétés syntaxiques et sémantiques et qu'ils sont interchangeables, ils n'ont aucun moyen de saisir le sens des symboles. On dit, techniquement, qu'ils ne sont pas *ancrés*, l'ancrage faisant référence à la capacité d'un modèle (ou d'une personne) à associer un mot à une réalité physique.

Pour bien comprendre cette notion et les problèmes qu'elle pose aux giga-modèles, Emily Bender et Alexander Koller, deux spécialistes du TAL, ont imaginé une expérience de pensée²³. Notons pour commencer que cette expérience a été imaginée en 2020, bien avant (à l'échelle de la recherche en TAL) le

développement de ChatGPT. Cette expérience met en scène trois protagonistes : une pieuvre incroyablement intelligente nommée Tentacool (un nom choisi par ChatGPT en lui demandant : *génère un nom de personnage d'une histoire d'enfant, ce personnage est une pieuvre*), ainsi que deux êtres humains, Alex et Jordan, résidant sur deux îles désertes reliées par un fil télégraphique leur permettant de communiquer.

Tentacool, qui réside dans les profondeurs de la mer, écoute les messages échangés entre Alex et Jordan. Mais, tapie au fond de l'océan, elle n'a jamais vu Alex, Jordan ou leurs îles et ignore même leur existence, se contentant d'observer les impulsions électriques qui transitent par le câble. Si elle ne connaît pas le langage humain, Tentacool excelle dans la détection de certaines régularités statistiques dans le signal. Après un certain temps, à force d'écouter de nombreuses conversations, elle parvient même à prédire avec une fiabilité surprenante le signal que Jordan générera en réponse aux messages d'Alex.

Un jour, Tentacool, lassée de sa solitude et de l'absence d'interaction, décide de couper le fil et de prendre la place d'Alex dans la conversation. Une question se pose alors : sera-t-elle capable de duper Jordan en poursuivant la conversation sans révéler que ce n'est plus Alex qui répond ? La réponse à cette question dépend largement du sujet abordé dans la conversation. Pour des discussions courantes, similaires à celles entre nos deux protagonistes, il ne devrait pas y avoir de problème. Tentacool peut, en effet, se contenter de sélectionner et de répéter les réponses adaptées qu'elle a mémorisées pendant sa phase d'observation. En revanche, imaginons que Jordan invente un nouveau moyen de communication plus efficace, en utilisant les noix de coco, abondantes sur l'île, et envoie les instructions de construction à Alex via le télégraphe, pour que ce dernier puisse construire ce nouvel appareil. Tentacool serait alors dans l'incapacité totale d'exécuter ces instructions. Quand bien même les fonds marins regorgeraient de noix de coco, elle ne saurait pas

mettre en correspondance le signal correspondant à « noix de coco » et l'objet physique à portée de ses tentacules. Les seules réponses que pourrait produire la pieuvre seraient des banalités (« Bonne idée ! », « Cool ! », ou encore « Pourquoi inventer un nouveau moyen de communication ? »).

Cette histoire illustre parfaitement un concept fondamental de la linguistique : la séparation du signifié (la forme sonore ou visuelle du mot) et du signifiant (le concept ou la signification que le mot évoque) identifiée par le linguiste suisse Ferdinand de Saussure dès 1916. On le retrouve dans l'argument de Noam Chomsky, qui affirmait que la capacité à parler correctement (aspect syntaxique) devait être distinguée de la capacité à donner du sens (aspect sémantique) : il tout à fait possible de construire une phrase syntaxiquement correcte qui n'a aucun sens, comme sa fameuse phrase « d'incolores idées vertes dorment furieusement ».

Ce conte renvoie également à une limite identifiée dès le début des travaux sur les modèles de langues : il existe une différence fondamentale entre la capacité à manipuler des symboles (les mots), comme le fait très bien Tentacool, et celle des giga-modèles à construire le sens porté par ces symboles, par exemple pour créer un objet à partir de sa description. En fin de compte, les modèles de langue ne sont que ce qu'Emily Bender, Timnit Gebru, Angelina McMillan-Major et Margaret Mitchell ont appelé des *perroquets stochastiques* (*stochastic parrots*) dans l'une de leurs critiques²⁴ les plus virulentes contre les modèles de langue (celle que nous avons déjà évoquée [ici](#)) : comme des perroquets, ils ne font que répéter mécaniquement des séquences de mots qu'ils ont appris à partir de leur corpus d'entraînement, sans véritable compréhension de leur contenu.

La personnification de ChatGPT renforce cette limite. Ses réponses sont, en effet, orientées pour satisfaire un certain anthropomorphisme, tout comme l'étaient celles d'Eliza, son lointain ancêtre : ChatGPT s'exprime à la première personne,

s'excusant en cas de malentendu (« je m'excuse de la confusion ») et adoptant parfois un ton flatteur envers l'utilisateur (« votre texte est bien écrit mais... »). Tout est orchestré pour créer l'illusion d'un échange avec un ami. C'est un choix délibéré des concepteurs de ChatGPT qui connaissent, sans aucun doute, les travaux des chercheurs spécialisés dans les interactions homme-machine qui réfléchissent depuis de nombreuses années à la meilleure manière d'échanger avec une IA²⁵.

Cependant, ce choix soulève divers problèmes, notamment en ce qui concerne la compréhension réelle de ChatGPT : il crée l'illusion que le chatbot comprend les requêtes, renforçant ainsi la confiance des utilisateurs envers ses réponses. Ainsi, lorsqu'on utilise ChatGPT comme un moteur de recherche, le système ne comprend pas la question qui lui est posée et se contente, pour répondre, de générer la séquence de mots jugée la plus probable en fonction de la suite de mots composant la question. Pourtant, tout dans l'interaction est fait pour donner l'impression inverse ! Les critiques soulignent donc à juste titre que la seule connaissance acquise par ChatGPT réside dans la distribution statistique des mots et que, bien que celle-ci puisse refléter une certaine représentation du monde, elle ne peut en aucun cas être assimilée à une *connaissance*.

Cette limite, repérée bien avant le début de la course aux grands modèles, ne remet nullement en question l'intérêt des giga-modèles. La capacité à manipuler correctement les symboles est, par exemple, tout à fait suffisante pour de nombreuses tâches et, notamment, toutes les tâches d'assistance à l'écriture. Mais rien n'empêche le modèle d'*halluciner*, c'est-à-dire de générer des textes syntaxiquement parfaits, mais dépourvus de sens ; seul un examen approfondi révélera qu'ils ont été inventés de toutes pièces et ne reposent sur aucun élément réel. Ainsi, quand on demande à ChatGPT de réaliser une synthèse sur un sujet scientifique (un état de l'art), il invente bien souvent des titres d'articles parfaitement plausibles et les utilise dans le contexte

approprié ; il va jusqu'à rédiger de faux articles scientifiques, et seule une lecture attentive par un spécialiste révélera la supercherie²⁶. De l'aveu même de Sam Altman²⁷, le président d'OpenAI : « C'est une erreur, pour le moment [décembre 2022], de se fier à ChatGPT pour quoi que ce soit d'important ! »

Ancrer les modèles de langue dans la réalité

Vu la rapidité de développement des giga-modèles, l'argument des perroquets stochastiques date de la préhistoire des modèles de langue, et les modèles actuels diffèrent fondamentalement de ceux qui existaient à l'époque où Emily Bender et ses collègues ont formulé cet argument. Tous les modèles (ou presque) intègrent désormais des jugements humains pendant leur phase d'apprentissage et sont affinés sur des corpus d'instructions. Ces étapes ont pour objectif premier d'aligner les textes générés sur les intentions des utilisateurs. Cependant, plusieurs expériences récentes suggèrent que ces nouvelles sources d'information semblent également ancrer les modèles de langage.

De manière intuitive, comme expliqué dans le chapitre 4 (voir [ici](#)), si nous prenons un modèle de langue *classique* (celui qui existait en 2020) et que nous remplaçons toutes les occurrences de « verte » par « anaérobique » dans le corpus d'apprentissage, ce modèle générera une phrase comme « l'aurore grelottante en robe rose et anaérobique » sans hésiter. Cependant, de telles phrases ne correspondraient pas aux intentions et aux jugements des utilisateurs et seraient donc pénalisées lors de l'apprentissage du modèle.

Cette intuition est confirmée par toute une série de travaux qui suggèrent que l'alignement des modèles avec les intentions humaines leur confère de nouvelles capacités. Par exemple, en analysant les représentations construites par un giga-modèle jouant à de jeux de plateau (typiquement aux échecs ou à Othello) et comment celles-ci évoluent au cours de la partie, des chercheurs

ont pu mettre en évidence que les représentations cachées du modèle contiennent des informations décrivant la position des pièces sur le plateau de jeu : ils ont même pu reconstruire la position des pièces directement à partir des représentations du réseau de neurones²⁸, une première étape vers l'ancrage des représentations dans la réalité.

D'autres expériences ont également montré la capacité des giga-modèles à suivre des entités. Imaginons une histoire pour enfants dans laquelle plusieurs personnages interagissent et se déplacent d'un endroit à un autre (par exemple, « Jordan va au salon, discute avec Alex avant de rejoindre Nour dans la cuisine. »). Après avoir *lu* une histoire (même plus compliquée que cet exemple), le modèle peut précisément indiquer l'emplacement de chaque personnage, démontrant ainsi avoir construit une représentation complète de l'histoire dans les différentes représentations cachées qu'il a construites.

Le plus surprenant dans toutes ces expériences est que si les modèles alignés sur les jugements humains ont des performances *correctes* sur ces tâches, les modèles entraînés uniquement à partir de la tâche de prédiction des mots masqués sont totalement incapables de fournir la moindre bonne réponse. La prise en compte des jugements humains fait donc émerger de nouvelles capacités dans les giga-modèles. Ainsi, le reproche souvent fait aux giga-modèles de n'être que corrélations entre mots est de moins en moins valide : comme le montraient déjà les expériences sur l'accord du verbe ([ici](#)), les giga-modèles sont tout à fait capables d'extraire l'essence de leurs entrées et d'en construire des représentations abstraites dépassant les formes brutes des mots. Toutes ces recherches sont toutefois récentes, et l'étude de l'impact de l'alignement des modèles de langage reste un sujet de recherche actif.

Vivre avec ChatGPT

Maintenant que nous avons bien compris le fonctionnement des giga-modèles et discuté certaines de leurs limites et des critiques qui leur sont régulièrement adressées, il est temps de nous interroger sur l'impact qu'ils auront sur notre quotidien.

La crainte pour l'emploi

La découverte par le grand public des giga-modèles et de leurs capacités étonnantes a suscité une vague d'inquiétude sans précédent parmi les cols blancs. Alors que l'automatisation et la robotisation des métiers manuels sont discutées depuis des années, le développement de programmes capables d'accomplir des tâches *intellectuelles* (rédaction de comptes rendus, d'analyses détaillées, de traduction ou de croisement d'informations) menace directement les emplois de catégories de personnes qui étaient convaincues qu'elles ne pourraient jamais être remplacées par des robots. Ainsi, une étude réalisée en octobre 2023 montrait que 42 % des Français ont peur de voir leur emploi supprimé à cause de l'IA^{[29](#)}.

Cette crainte semble confirmée par les premières annonces de licenciements attribués à l'intervention de l'IA. En juin 2023, le quotidien allemand *Bild* a déclaré, par exemple, vouloir se séparer d'un tiers de ses employés pour les remplacer par des outils d'IA^{[30](#)}. Bien que l'établissement d'un lien de causalité directe entre l'IA et les licenciements soit sujet à caution, il est certain que de nombreux métiers sont ou seront complètement bouleversés par les giga-modèles. Selon le Forum économique mondial, une organisation de lobbying internationale qui organise des réunions annuelles de dirigeants mondiaux et de chefs d'entreprise, dont le fameux forum de Davos, l'IA devrait détruire 14 millions d'emplois au cours des prochaines années, obligeant 25 % des salariés à se reconverter. La banque d'affaires américaine Goldman Sachs considère, quant à elle, que la moitié des salariés verront

leur emploi exposé à un certain degré d'automatisation par l'IA, cette proportion atteignant 80 % aux États-Unis.

Il faut toutefois se rappeler que les préoccupations liées à l'automatisation du travail et, en particulier, à son impact sur l'emploi, sont présentes depuis longtemps. Cette « anxiété causée par l'automatisation » (*automation anxiety*³¹) est même un sujet d'étude pour les sociologues et les économistes qui se demandaient dès 2015 : « pourquoi y a-t-il encore autant d'emplois ? »³². Ces questions ont naturellement émergé dès les débuts de l'IA et reviennent régulièrement sur le devant de la scène. Par exemple, en 2018, le PDG d'Alphabet, Sundar Pichai, a présenté une démonstration d'un nouveau service, Google Assistant, capable grâce aux progrès d'Alphabet en TAL, en apprentissage profond et plus généralement en IA, de passer un appel téléphonique (littéralement) à un restaurateur pour faire une réservation ou à un coiffeur pour prendre un rendez-vous. Aussitôt, de nombreux spécialistes ont prédit une véritable révolution et la disparition de nombreux emplois, notamment dans les centres d'appels. Plus de 6 ans après, la révolution annoncée semble loin de s'être concrétisée et Alphabet a même réduit ses ambitions en ne proposant plus ce service que sur les téléphones mobiles.

Il est difficile de prévoir l'impact du TAL et de l'IA sur l'emploi. Mais il ne faut pas oublier que depuis de nombreuses années, des outils de TAL sont déployés et utilisés quotidiennement par des millions de personnes : filtres anti-spam, correcteurs orthographiques, moteurs de recherche, systèmes de traduction automatique, filtrage d'articles, etc. Tous ces outils reposent sur la capacité des ordinateurs à comprendre et à manipuler du texte et sont parfaitement intégrés à notre quotidien.

Plutôt que de se lancer dans des prévisions sans fondement, il est plus sage d'examiner ce qui s'est déjà produit dans un domaine qui a été profondément affecté par le développement du TAL : la

traduction. Il y a de nombreuses années que l'on annonce la fin de la traduction humaine, supposée être remplacée par sa version automatique, plus rapide et moins coûteuse. Dès 2012, Google Translate traduisait déjà l'équivalent d'un million de livres chaque jour, soit l'équivalent de la production annuelle des traducteurs du monde entier.

Malgré l'amélioration constante des systèmes de traduction automatique et leur usage croissant, les traducteurs continuent à travailler. Leur métier a, bien sûr, profondément évolué avec le développement de la traduction automatique (et du TAL en général) et repose aujourd'hui sur des outils informatiques, tels que les concordanciers (permettant de visualiser comment un mot est utilisé dans une langue donnée, c'est-à-dire dans quel contexte) ou les mémoires de traduction (permettant de stocker et de réutiliser facilement des traductions déjà effectuées), voire directement des systèmes de traduction automatique : de plus en plus souvent, la traduction automatique est vue comme un premier jet que les traducteurs devront corriger (techniquement *post-éditer*) pour en rectifier les erreurs, garantir la cohérence du style et le respect des consignes d'écriture de l'entreprise³³, et surtout certifier que la traduction est correcte et peut être diffusée.

Le développement de la traduction automatique n'a donc pas tué (et ne tuera pas) la *biotraduction*, mais a profondément modifié la manière dont les traducteurs travaillent tout en créant de nouveaux métiers. Elle a également ouvert de nouveaux usages à la traduction : les utilisateurs de service de traduction traduisent des documents qu'ils n'auraient jamais envoyés à des traducteurs professionnels, soit parce qu'ils n'en ont pas les moyens, soit parce que la qualité demandée ne justifie pas l'intervention d'un humain (c'est typiquement le cas lorsque l'on traduit un e-mail pour s'assurer de bien le comprendre).

L'enseignement de la traduction a également profondément évolué : toutes les formations en traduction incluent désormais

des cours sur les principaux outils informatiques et forment leurs étudiants à la post-édition. De manière intéressante, l'expérience montre que pour enseigner la post-édition, il est nécessaire de commencer par la traduction manuelle : la maîtrise des nouveaux usages nécessite une bonne connaissance des anciennes manières de traduire et des problématiques qui y sont associées.

Cette vision peut sembler optimiste et correspondre uniquement à celle d'un chercheur en TAL. Il ne faut, en effet, pas oublier que le développement de la traduction automatique a également entraîné une pression accrue sur les traducteurs, notamment sur leurs salaires : le coût de la post-édition étant généralement compris entre 50 % et 70 % du coût d'une traduction complète, alors que cette étape n'est pas forcément plus rapide ni plus facile.

À la fin de l'année 2023, les premières études scientifiques sur l'impact de ChatGPT sur l'emploi ont commencé à être publiées, offrant une image plus précise de l'influence des giga-modèles sur le monde du travail. Des chercheurs de Harvard, par exemple, ont mesuré³⁴ comment ChatGPT pouvait aider les consultants du Boston Consulting Group, l'un des principaux cabinets de conseil en stratégie, dans des tâches qualifiées de « réalistes, complexes et nécessitant une utilisation intensive de connaissances ». Leurs résultats sont sans équivoque : l'utilisation de ChatGPT permet d'accomplir davantage de tâches (une augmentation de 12,2 % pour être précis), de manière plus rapide (un gain de temps estimé à 25 %) tout en améliorant la qualité (une amélioration de 40 %). Il y a toutefois un point rassurant : sur les tâches les plus difficiles, ChatGPT a eu un impact négatif sur le travail des consultants. Il est donc important de bien connaître les limites de l'outil pour pouvoir choisir quand l'utiliser ou non.

Comme on pouvait s'y attendre, les chercheurs ont également mis en évidence que l'IA générative est de plus en plus souvent utilisée à la place des humains. Une autre étude³⁵ basée sur l'analyse des contrats passés sur Upwork, l'une des principales

plateformes de freelance, montre que les concepteurs-rédacteurs et les graphistes indépendants trouvant leurs missions sur cette plateforme ont obtenu en moyenne 2 % de contrats en moins, et leur revenu mensuel a baissé de 5 % depuis l'introduction de ces modèles.

Ces premières observations doivent être interprétées avec prudence, car les données disponibles sont encore limitées. Il est difficile de déterminer si les tendances observées ne sont que des effets à court terme, potentiellement liés à un effet de mode, ou si elles traduisent une tendance globale. Elles sont toutefois en parfait accord avec les prévisions de nombreux acteurs et correspondent à ce qui a été observé dans les métiers de la traduction. Les nombreuses craintes exprimées quant à l'impact sur l'emploi semblent donc parfaitement justifiées. Pourtant, si l'on examine le cas des programmeurs et des scénaristes, deux professions jugées par beaucoup comme directement menacées par ChatGPT, les choses ne sont pas nécessairement aussi sombres.

Comme nous l'avons mentionné, ChatGPT est très bon pour générer du code à partir d'une description textuelle. Des outils plus spécialisés comme Copilot (un ChatGPT spécialisé dans la génération automatique de code, également développé par OpenAI en partenariat avec GitHub, un outil de gestion de code de Microsoft) existaient avant la sortie de ChatGPT et avaient même été intégrés dans les outils de développement de Microsoft. Ces outils sont capables de traduire une description d'un problème en langage naturel en code (et vice-versa), une menace directe au travail des programmeurs.

Il y a toutefois un gouffre qui sépare le métier de programmeur et les capacités de ChatGPT, et que les giga-modèles sachent écrire quelques lignes de code ne doit tromper personne : la plus grande partie du travail d'un informaticien consiste à comprendre le problème qu'il doit traiter (les besoins du client) et la base de code

existante pour voir comment la modifier ou y intégrer la nouvelle fonctionnalité, et surtout vérifier que les modifications apportées ne *cassent* pas le reste du programme.

Exactement comme pour la traduction, l'IA est donc un nouvel outil qui permettra aux informaticiens de travailler plus rapidement en se concentrant sur la partie difficile de leur travail, et il est peu probable qu'à moyen terme, du moins, les informaticiens soient remplacés par une machine. Il est en revanche tout à fait envisageable que les outils d'IA démocratisent l'accès à la programmation : une personne sans aucune compétence en informatique pourra demander à ChatGPT (ou tout autre service) de lui écrire directement un morceau de code automatisant son travail. Qui n'a jamais eu l'impression de perdre son temps en effectuant une tâche répétitive et en rêvant de la confier à une machine ?

Parmi les métiers que beaucoup jugent directement menacés par ChatGPT et les giga-modèles, on trouve également tous les métiers créatifs. Les progrès de l'IA ont, par exemple, joué un rôle certain dans la grève des scénaristes et des acteurs³⁶ américains qui a paralysé Hollywood une bonne partie de l'année 2023. Une des principales revendications des grévistes était de limiter les recours à l'IA dans la profession. En effet, l'écriture du scénario, comme toutes les tâches d'écriture, peut être grandement facilitée par l'utilisation d'un giga-modèle comme assistant d'écriture, que ce soit pour corriger l'orthographe ou le style, compresser ou, au contraire, allonger le texte, etc. Il est même envisageable d'automatiser l'écriture de certaines scènes, par exemple, en fournissant une amorce comme : « Peux-tu m'écrire un dialogue entre deux amies se racontant leurs souvenirs de vacances à la plage avec une quinzaine de répliques ».

Mais si ChatGPT est capable d'écrire des scènes, ce n'est qu'en réponse à une demande précise, et encore, ces scènes sont très souvent des combinaisons et des adaptations de scènes existantes.

La situation est exactement la même que pour les images générées par DALL·E : celles-ci donnent souvent l'impression d'être originales, mais de nombreux artistes y ont reconnu des morceaux de leurs œuvres, motivant plusieurs dépôts de plainte. Ce n'est pas une surprise quand on sait comment fonctionnent ces modèles ! Une fois de plus, il est probable que pendant de longues années encore, les giga-modèles restent des outils que les scénaristes devront apprendre à maîtriser et que, même s'ils changent fondamentalement leur manière de travailler, ils ne les remplaceront pas.

La perte de la vérité

En plus des craintes (légitimes) concernant l'emploi, une autre préoccupation s'est très vite répandue, cette fois liée à la possibilité d'utiliser ChatGPT, et plus généralement les outils d'IA générative, dans des campagnes de désinformation. Pour reprendre les mots de Geoffrey Hinton, l'un des pères de l'apprentissage profond, qui a décidé de quitter son poste chez Alphabet pour pouvoir critiquer librement les derniers développements des giga-modèles, le risque est que « les gens normaux ne puissent plus distinguer le vrai du faux »³⁷.

Il est vrai que pendant longtemps, la capacité à rédiger un texte correctement était considérée comme une preuve d'éducation, voire d'intelligence. Il suffit de se souvenir de la place qu'occupent les dissertations à l'école et de l'importance accordée à l'enseignement de l'orthographe. Il ne fait aucun doute qu'un avis sur un restaurant truffé de fautes d'orthographe ou de grammaire, ou un article de journal écrit maladroitement, seront rarement pris au sérieux et leur contenu sera souvent ignoré.

Les giga-modèles changent fondamentalement la donne dans ce domaine en offrant, comme le montre la figure ci-dessous, la possibilité de générer un grand nombre de textes donnant l'impression d'avoir été écrits par une personne et donc d'être le

fruit d'une réflexion avancée. Quand on sait, comme l'explique parfaitement David Chavalarias dans son ouvrage *Toxic Data*, qu'il suffit de poster suffisamment de messages crédibles bien ciblés pour faire basculer une élection ou transformer une manifestation en bataille rangée, les enjeux apparaissent clairement.

Personne : peux-tu me générer 3 avis (un paragraphe chacun) sur une pizzeria de Chaumont, la « villa romana ». J'ai adoré la qualité des pizzas, les serveurs étaient vraiment sympas et en plus la carte était originale

ChatGPT : Avis 1: La Villa Romana à Chaumont est une véritable pépite pour les amateurs de pizzas exquises. J'ai récemment eu l'occasion de savourer leurs créations et je suis totalement conquis par la qualité exceptionnelle de leurs pizzas. La pâte est parfaitement cuite, croustillante à l'extérieur et moelleuse à l'intérieur, tandis que les garnitures sont fraîches et généreuses. Le personnel, d'une amabilité remarquable, a contribué à rendre l'expérience encore plus agréable. Leur service attentionné et souriant a ajouté une touche chaleureuse à cette expérience culinaire mémorable.

Avis 2: Une soirée à la Villa Romana à Chaumont est une véritable escapade gourmande. Ce qui m'a particulièrement séduit, outre la délicieuse pizza, c'est l'ambiance conviviale qui règne dans cet établissement. Les serveurs sont non seulement professionnels, mais aussi incroyablement sympathiques. Ils ont su créer une atmosphère accueillante qui vous fait sentir comme chez vous. De plus, la carte de la pizzeria est tout simplement originale, proposant des combinaisons de saveurs inédites et surprenantes qui témoignent de la créativité du chef. Une adresse à recommander sans hésitation pour une soirée délicieusement différente.

Avis 3: La Villa Romana à Chaumont est une adresse que je recommande vivement aux amateurs de pizzas à la recherche d'une expérience gustative exceptionnelle. Les pizzas sont préparées avec une attention méticuleuse aux détails, et chaque bouchée est une explosion de saveurs. Ce qui distingue également cet endroit, c'est l'attitude exceptionnelle du personnel. Les serveurs sont non seulement efficaces dans leur service, mais ils le font avec un sourire et une gentillesse qui ajoutent

indéniablement à l'agrément de la soirée. De plus, la carte originale offre un choix diversifié qui saura satisfaire tous les goûts. Une expérience culinaire mémorable à la Villa Romana.

Figure 18. Exemples d'avis sur un restaurant générés par ChatGPT

Il ne fait également aucun doute que les giga-modèles augmentent également les risques de fraude aux examens, non pas tant en facilitant l'accès à des informations qui sont bien souvent consultables en ligne, qu'en reformulant les informations extraites pour répondre exactement à la question, rendant très difficile la détection du plagiat (qui va généralement chercher à identifier des morceaux de textes identiques). Les giga-modèles retrouveront non seulement les réponses, mais sauront les reformuler dans le style adéquat pour convaincre l'examineur. Si pour beaucoup d'enseignants, le développement de ces modèles offre de nombreuses opportunités pour améliorer l'accès aux connaissances et développer l'esprit critique des étudiants (par exemple en leur demandant de critiquer les réponses de ChatGPT), la tentation est grande d'interdire l'utilisation de ce logiciel afin d'éviter de remettre en cause les méthodes d'évaluation habituelles. Il n'y a pour le moment qu'une seule certitude : le développement de l'IA bouleverse aussi bien la manière d'enseigner que l'évaluation des étudiants et les conférences, projets de recherche, réflexions, etc.

Pour bien comprendre cet impact des giga-modèles sur notre quotidien, il faut se rendre compte qu'il n'existe actuellement aucun moyen de savoir si la paternité d'un texte revient à un giga-modèle ou à un être humain. Ainsi, si je reçois un poème comportant une véritable déclaration d'amour, rien ne me permettra de savoir s'il a été généré en quelques secondes par un giga-modèle à partir de quelques indications personnelles (parce que, oui, ChatGPT sait parfaitement écrire des poèmes respectant les contraintes de forme !), ou s'il exprime la passion d'une personne assez motivée pour consacrer le temps nécessaire à la

recherche de formules marquantes et du respect des contraintes de forme imposées par le genre. Là réside toute la difficulté soulevée par le développement des giga-modèles.

Les capacités des giga-modèles comme aide à la rédaction nécessitent donc de remettre en question toutes nos échelles de valeur en ce qui concerne l'évaluation de la fiabilité de certaines informations. Ainsi, pendant longtemps, une lettre de motivation bien rédigée était une preuve convaincante de l'enthousiasme et de la qualité de la personne. Cependant, il est probable qu'à très court terme, toutes les lettres de motivation soient corrigées voire rédigées directement par des giga-modèles, les rendant ainsi toutes impeccables sur le plan formel. Par conséquent, il sera nécessaire, pour évaluer la personne, de prêter une attention accrue au contenu du courrier. Ironiquement, il est tout aussi probable que les personnes chargées de lire les lettres de recommandation demandent de plus en plus souvent aux mêmes giga-modèles de les résumer, voire de les jauger directement. Il ne fait donc aucun doute que le développement des giga-modèles et leur utilisation qui ne sera que de plus en plus fréquente nécessite de remettre en cause de nombreuses habitudes dans notre évaluation des informations accessibles.

Réguler l'IA et les giga-modèles

La remise en question de la vérité n'est qu'une des craintes soulevées par le développement de l'IA. Ces dernières années, de nombreux scientifiques, ainsi que des chefs d'entreprise ayant joué un rôle majeur dans le développement de cette technologie, ont mis en garde contre les risques liés aux nouvelles capacités de l'IA. Les termes employés sont particulièrement alarmistes : selon Elon Musk, l'IA serait « plus dangereuse que les bombes nucléaires »³⁸, et selon un appel du Center for AI Safety³⁹, elle présenterait un « risque d'extinction pour l'humanité ». Le ton de ces craintes est tellement alarmiste que l'on pourrait presque

croire à une nouvelle théorie conspirationniste. Mais les signataires de cet appel sont tous des spécialistes de l'IA de premier plan, dont Yoshua Bengio et Geoffrey Hinton, deux des pères de l'apprentissage profond récompensés par le prix Turing pour leurs travaux dans ce domaine, les fondateurs et les principaux cadres d'OpenAI (Sam Altman, Ilya Sutskever, Mira Murati, ...), Demis Hassabis le PDG de DeepMind (l'entreprise qui a développé AlphaGo), et bien d'autres scientifiques de premier plan.

Il est vrai que, comme nous l'avons décrit dans les chapitres précédents, l'IA a réalisé, ces dernières années, des progrès spectaculaires et exponentiels, et pas seulement dans le domaine du langage. Vu la vitesse à laquelle les ordinateurs sont devenus capables de parler (avec les limites que nous avons décrites dans ce chapitre), et surtout l'étendue des possibilités et des applications que les derniers modèles ouvrent, on peut se demander jusqu'où pourra aller l'IA. De nombreux chercheurs anticipent que les progrès récents vont continuer à se poursuivre à la même vitesse, aboutissant au développement de ce que le philosophe Nick Bostrom appelle une « super-intelligence »⁴⁰ dont les capacités dépasseraient largement celles des humains, ouvrant la possibilité de réaliser de nombreux rêves (et cauchemars) de la science-fiction.

Ces craintes restent pour le moment très spéculatives et fondées essentiellement sur des conjectures. Les capacités et les limites des modèles d'IA, notamment des modèles génératifs, sont bien identifiées, et aucun signe concret du développement d'une super-intelligence n'est encore apparu. Cependant, l'histoire récente du domaine nous apprend que des choses qui semblaient inconcevables aux chercheurs il y a quelques années encore (voire quelques mois) étaient finalement réalisables avec des modèles relativement simples. Qui sait ce que l'avenir nous réserve ! Il ne faut toutefois pas que ces risques hypothétiques à long terme nous fassent perdre de vue que de nombreuses méthodes d'IA sont

déjà déployées à grande échelle : les discussions sur ce qui va se passer à long terme ne devraient pas éclipser celles sur les conséquences immédiates de l'utilisation massive de méthodes d'IA, par exemple les problèmes liés aux droits d'auteur ou à l'exploitation de travailleurs déjà mentionnés ou ceux liés à la reproduction de certains biais des données d'apprentissage (notamment en termes de couleur de peau ou de genre⁴¹).

Toutes ces questions ont divisé les chercheurs en IA en deux camps : les *decels*, partisans d'un ralentissement des recherches dans le domaine, notamment pour laisser le temps d'analyser l'impact des progrès sur la société, et les *accels* qui, au contraire, essaient d'accélérer le développement de l'IA et de lancer autant de produits que possible le plus vite possible. Les craintes de ce premier camp ont trouvé un écho dans le monde politique. De nombreux pays, allant de la Chine aux États-Unis en passant par l'Union européenne, ont mis en place des mesures de régulation de l'IA. Le Parlement européen a adopté, le 14 juin 2023, un premier texte régulant l'IA, (une « proposition de négociation » dans le processus législatif européen) dont l'objectif principal est à la fois de garantir les droits fondamentaux et les valeurs de l'Union européenne, tout en mettant en place un cadre juridique stable facilitant l'innovation et les investissements. Cet *AI Act*, en discussion depuis avril 2021, définit une « pyramide des risques » permettant de décrire les systèmes d'IA. Celle-ci est composée de quatre niveaux allant de « minime » à « inacceptable » et acte l'interdiction des systèmes entrant dans cette dernière catégorie⁴².

Les giga-modèles sont identifiés comme présentant un risque de niveau 2, au même niveau que les systèmes de génération ou de manipulation des images et de l'audio, connus sous le nom de *deep fake*, avec un risque qualifié de « limité ». Les systèmes d'IA de niveau 2 ne seront pas soumis à une évaluation préalable à leur commercialisation (contrairement aux IA de niveau 3, jugées « à haut risque »), mais auront une obligation de transparence. Leur exploitant devra expliquer leur fonctionnement et ces systèmes

devront se conformer à trois exigences : leurs sorties devront faire apparaître le fait qu'elles ont été générées par une IA, les modèles devront empêcher, par leur conception, de générer du contenu illégal, et les données protégées par le droit d'auteur utilisées pour leur apprentissage devront être clairement identifiées.

L'IA Act veut également protéger les titulaires des droits d'auteur (journalistes, écrivains, artistes, etc.) en obligeant les créateurs de modèles à publier des informations suffisamment précises sur les données qu'ils utilisent. L'objectif est d'identifier les ayants-droits et de leur permettre de demander une rétribution. Cette perspective suscite des inquiétudes au sein des entreprises d'IA. Le PDG de Mistral AI a, par exemple, déclaré qu'« une transparence complète des données d'entraînement, notamment des URL, reviendrait à signer la mort de l'entreprise ». Il a été suivi par le Président de la République Emmanuel Macron, qui, depuis l'émergence de potentiels champions français, s'oppose à toute idée d'une régulation contraignante en Europe. Les négociations autour de l'IA Act sont loin d'être terminées⁴³ !

Aux États-Unis, le Président Joe Biden a marqué un tournant dans l'histoire du pays en signant, le 30 octobre 2023, un décret visant également à réguler l'IA. C'est une première pour ce pays, qui a une longue tradition de non-régulation de la technologie. Ce décret impose aux entreprises de tester la sécurité de leurs modèles et de partager les résultats avec une autorité de régulation gouvernementale. De plus, il établit des normes visant à détecter les contenus générés par des IA afin de prévenir la propagation des fausses informations.

Si la mise en œuvre de ces textes fait encore face à de nombreux obstacles, tant politiques (des pourparlers avec les pays de l'Union européenne au sein du Conseil européen sur la forme finale de la loi étaient toujours en cours fin 2023) que techniques (on ne sait, par exemple, pas identifier le contenu produit par un modèle génératif à l'heure actuelle), ces deux initiatives illustrent

parfaitement la prise de conscience de l'impact que l'IA sur notre quotidien et des risques liés à son développement. Il n'existe toutefois pas de consensus sur la solution à apporter à ces problèmes : à la fin de l'année 2023, la bataille faisait toujours rage entre les partisans d'une régulation et ceux qui craignaient que celle-ci ne freine l'innovation, compromettant ainsi les promesses de croissance économique et d'amélioration de la qualité de vie apportées par les giga-modèles et l'IA.

Cette opposition était particulièrement visible lors du premier sommet sur les risques associés à l'IA organisé à Londres les 1^{er} et 2 novembre 2023. À l'initiative de Rishi Sunak, le premier ministre anglais, cette réunion a rassemblé les représentants de 28 États et un large éventail d'entreprises travaillant autour de l'IA, tant de grandes multinationales (Alphabet, OpenAI, Meta, etc.) que des startups spécialisées dans le développement de giga-modèles (MistralAI, Conjecture, etc.). Les retombées sont pourtant maigres et cet événement a surtout mis en évidence les divergences d'intérêts entre les participants, qu'il s'agisse des pays (cherchant à protéger leurs empires numériques et leurs fleurons technologiques) ou des entreprises, chacune défendant un modèle économique et une régulation conformes à ses intérêts.

Le débat autour de la question des risques de l'IA a même ébranlé OpenAI. Le 17 novembre 2023, Sam Altman, le PDG d'OpenAI devenu l'une des figures emblématiques de l'IA générative grâce au succès de ChatGPT, a été limogé de manière totalement inattendue par le conseil d'administration de l'entreprise qu'il avait fondée. Celui-ci lui reprochait d'orienter trop fortement la stratégie d'OpenAI vers le succès commercial à court terme, sans prendre en compte les risques associés au développement d'une IA de plus en plus *intelligente*. Rappelons qu'après le succès de ChatGPT, OpenAI a établi un partenariat privilégié avec Microsoft, qui a investi plus de 10 milliards de dollars dans l'entreprise mais que, malgré cet investissement, les frais de fonctionnement de ChatGPT rendent l'entreprise fortement

déficitaire. Après de multiples rebondissements dignes d'un feuilleton télévisé, culminant avec l'annonce de son embauche chez Microsoft et la menace de 700 des 770 employés d'OpenAI de le rejoindre chez le géant de l'informatique, il a été réintégré le 22 novembre dans ses fonctions, après avoir obtenu la nomination d'un nouveau conseil d'administration.

La crise traversée par OpenAI met en lumière un dilemme auquel de nombreuses entreprises développant des giga-modèles sont actuellement confrontées : trouver un équilibre entre la génération des revenus nécessaires pour maintenir l'infrastructure utilisée pour l'apprentissage et l'utilisation de ces modèles, et la publication de nouveaux modèles prenant en compte l'ensemble des risques soulevés par ceux-ci. La recherche de cet équilibre, potentiellement perturbée par la mise en place de régulations contraignantes, va très certainement impacter fortement le développement des giga-modèles ces prochaines années. L'histoire n'est pas finie !

ÉPILOGUE

Nous voilà au terme de l'histoire de la révolution ChatGPT (et des giga-modèles), même si celle-ci est loin d'être terminée. Si l'ambition de doter les ordinateurs du langage est presque aussi vieille que l'informatique, de multiples voies d'accès ont été tentées pour atteindre ce graal. Au fil des explications, j'espère vous avoir convaincu de trois choses :

- Les giga-modèles ont des capacités inimaginables qui vont (et qui ont déjà) profondément changer notre quotidien, notamment notre rapport au langage.
- Leurs capacités reposent non pas sur les progrès continus d'une technologie, mais sur une véritable révolution : ils possèdent des capacités fondamentalement différentes de celles des premiers systèmes de TAL, et même des premiers modèles de langue, notamment celle d'ancrer leurs connaissances et leurs représentations.
- La mise au point de ces modèles repose essentiellement sur une longue succession d'essais et d'erreurs qui a été guidée principalement par des contraintes techniques (le besoin de disposer d'algorithmes pouvant s'exécuter facilement sur des GPUs pour simplifier), et non sur une véritable compréhension ni du fonctionnement du cerveau ni de la manière dont les humains comprennent et manient le langage.

Si les giga-modèles sont aujourd'hui sur le devant de la scène, ils demeurent très mystérieux. Se hasarder à livrer des

prévisions sur leur futur, aussi bien technique et scientifique que socio-économique, reste une véritable gageure.

Puisqu'ils ont été mis au point par tâtonnement, personne ne sait aujourd'hui parfaitement comment ils fonctionnent ni comment ils acquièrent toutes leurs fascinantes capacités. L'étude de ces modèles et de leurs capacités est un sujet de recherche actif, et de nombreuses questions fondamentales restent encore sans réponse : que peuvent faire les giga-modèles ? Comment fonctionnent-ils ? À quel point sont-ils bons dans ce qu'ils font ? Est-ce que l'intelligence n'est, finalement, qu'un très grand nombre de combinaisons linéaires, une opération tellement simple ? On peut parier qu'en améliorant notre compréhension de leur fonctionnement, mais aussi celle de la façon dont le cerveau représente et utilise le langage, de nouvelles architectures plus économes en données et nécessitant moins de paramètres (et donc de puissance de calcul) verront le jour.

D'un point de vue socio-économique, l'engouement que les giga-modèles ont suscité est sans précédent, à la mesure des perspectives et des révolutions qu'ils rendent possibles. Pourtant, de très nombreux problèmes restent à résoudre, liés notamment au respect des droits d'auteur et des données personnelles. Un nombre croissant d'analystes s'interrogent également sur les sommes colossales investies dans ces modèles (aussi bien pour utiliser les modèles existants que pour la recherche qu'ils exigent en amont) et prévoient un ralentissement des investissements, voire un nouvel hiver de l'IA¹.

Plus fondamentalement, la capacité maintenant avérée des ordinateurs à écrire du texte aussi bien que les humains nous contraint à remettre en cause de nombreuses habitudes, et soulève bien des questions.

Les annonces de progrès continuent de se succéder à un rythme effréné, entretenant un certain effet de mode autour de l'IA et des giga-modèles, et servant surtout aux entreprises à justifier, aux yeux du grand public et de leurs investisseurs, leur maîtrise d'une technologie d'une grande valeur. Même si elle ne bénéficie ni des mêmes moyens ni de la même publicité, une recherche plus fondamentale continue de se développer autour de la conception, de l'analyse et de la compréhension des giga-modèles, mais aussi sur l'impact des développements de ces modèles. Nous ne sommes, finalement, qu'au début de l'aventure. Les pages les plus passionnantes sont encore à écrire.

NOTES ET RÉFÉRENCES

Prologue

1. « *Dave, stop... Stop, will you? I'm afraid, Dave... Dave... my mind is going... I can feel it.* »

2. L'informatique décisionnelle (*business intelligence*) a pour objectif de collecter, d'analyser et de restituer toutes les informations présentes dans le système d'information d'une entreprise (celles stockées dans un format numérique) afin d'offrir à l'équipe dirigeante une vision générale de l'entreprise et de l'aider dans ses décisions stratégiques.

3. <https://www.domo.com/data-never-sleeps>

4. Voir, par exemple, <https://www.petitpetitgamin.com/2013/03/05/50-erreurs-de-traduction/>

5. « Facebook blames 'technical error' for Xi Jinping offensive name translation gaffe », *The Guardian*, 19 janvier 2020.

6. Hern A., « Facebook translates 'good morning' into 'attack them', leading to arrest », *The Guardian*, 24 octobre 2017.

7. Milmo D., « ChatGPT reaches 100 million users two months after launch », *The Guardian*, 2 février 2023.

8. <https://www.loue.fr/article/Et-ça-il-sait-le-faire-ChatGPT-La-nouvelle-affiche-estivale-des-Fermiers-de-Loué/a210.aspx>

9. Alphabet est un conglomérat d'entreprises qui regroupe notamment Google, YouTube, Gmail et plusieurs laboratoires spécialisés en IA : DeepMind, Google Brain, X, etc.

10. <https://www.economie.gouv.fr/strategie-nationale-intelligence-artificielle>

11. VivaTech est un salon annuel de l'innovation technologique à Paris qui réunit des startups, des entreprises et des leaders de l'industrie pour partager des idées, présenter des innovations et favoriser la collaboration.

[12.](#) Bergounhox G., « La start-up française Mistral AI lève 105 millions d'euros pour développer des modèles d'IA générative », *L'Usine digitale*, 23 juin 2023.

Des programmes pour parler, un défi insurmontable

[1.](#) Le prix Turing décerné par l'ACM, une association internationale visant à développer et soutenir la recherche scientifique et l'innovation informatique, est considéré comme l'équivalent du prix Nobel en informatique.

[2.](#) Cité par Daniel Crevier dans son ouvrage *À la recherche de l'intelligence artificielle*, Champs Flammarion, 1997.

[3.](#) Eliza parlait en anglais, ce qui facilite grandement l'écriture des règles de transformation : il n'y a, par exemple, pas besoin de changer la conjugaison des verbes comme cela aurait été le cas si elle parlait en français.

[4.](#) « *I had not realized ... that extremely short exposures to a relatively simple computer program could induce powerful delusional thinking in quite normal people.* »

[5.](#) *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, Basic Books, 1996.

[6.](#) L'ARPA, devenue aujourd'hui la *Defense Advanced Research Projects Agency* (DARPA), est une agence américaine chargée de la recherche et développement des nouvelles technologies destinées à un usage militaire. Elle joue un rôle central dans le développement de la recherche en IA depuis la naissance du domaine et a financé une partie du développement de presque toutes les applications majeures du TAL, y compris la traduction automatique, la recherche d'informations, la reconnaissance vocale, l'extraction d'informations, le résumé de texte, l'analyse des sentiments et les systèmes de dialogue.

[7.](#) Par soucis de simplicité et de clarté, je parlerai essentiellement des textes écrits, même si la communication orale joue un rôle essentiel dans la communication entre les humains et entre humains et ordinateurs. Les problèmes rencontrés dans la compréhension de l'oral sont les mêmes que pour la compréhension de l'écrit.

[8.](#) Les phrases sont, par définition, des ensembles syntaxiques autonomes qui expriment une idée *complète*. Afin de limiter le nombre de mots à analyser et donc le temps de calcul, une chaîne de traitement se concentre généralement sur une seule phrase sans prendre en compte le contexte dans lequel celle-ci apparaît.

9. Il existe plusieurs « jeux d'étiquettes » (*tagset*) pouvant décrire les catégories grammaticales avec plus ou moins de précision. Un des jeux les plus utilisés aujourd'hui est le *Universal Dependencies*, dont une description complète se trouve sur <https://universaldependencies.org/>

10. Une langue qualifiée d'« agglutinante » forme certaines constructions grammaticales en ajoutant des suffixes et des préfixes à des noms. Par exemple, en finnois, le mot « *taloissani* » est composé de *talo*, le nom désignant une maison, et de trois suffixes : « *-i-* » la marque du pluriel, « *-ssa-* » l'équivalent de « dans » (techniquement, une marque de l'inessif) et « *-ni* » la marque du possessif. Ce mot correspond donc à trois mots français (« dans ma maison »). De manière similaire, en néerlandais, il est possible de créer facilement des mots par compositions : le mot « *meervoudigepersoonlijkheidsstoornis* » se traduit en français par « trouble de la personnalité multiple ».

11. Ils sont regroupés sur un site de la Direction de l'information légale et administrative : <https://www.vie-publique.fr/discours-dans-lactualite/269998-les-voeux-des-presidents-de-la-republique-depuis-1974>

12. Cette intuition est formalisée par une loi empirique, la loi de Zipf, qui stipule que la fréquence d'occurrence d'un mot est inversement proportionnelle à son rang dans une liste triée par fréquence : le mot le plus fréquent (rang 1) apparaît deux fois plus souvent que le deuxième mot le plus fréquent (rang 2), le troisième mot le plus fréquent apparaît trois fois moins souvent que le deuxième mot le plus fréquent, et ainsi de suite.

13. De nombreuses manières de définir et de combiner le *tf* et l'*idf* ont été proposées au fil des années. Aujourd'hui, la plupart des applications considèrent une définition un peu plus compliquée que celle donnée dans le texte : l'*idf* d'un mot apparaissant dans n documents différents est défini par : $\frac{N + 1}{n + 1}$. Le *tf* et l'*idf* sont généralement combiné de la manière suivante : $tf \times (\log idf + 1)$

14. Pour déterminer la pertinence d'une page, les moteurs de recherche considèrent de nombreuses informations que l'on peut regrouper en deux grandes catégories : les informations visant à déterminer la qualité de la page (date de mise à jour, nombre de liens entrants et sortants, etc.) et les informations mesurant l'adéquation entre la requête et le contenu du document (principalement le score *tf.idf*).

Apprendre en faisant : la révolution de l'apprentissage statistique

1. Il existe, en toute rigueur, deux types d'apprentissage automatique (*machine learning*) : l'apprentissage symbolique et l'apprentissage statistique. L'apprentissage statistique est aujourd'hui omniprésent, tandis que les méthodes symboliques ont quasiment disparu. Par soucis de simplicité, j'utiliserai, dans la suite du texte, le terme « apprentissage » pour désigner la voie statistique.

2. Le « *show don't tell* » est une technique d'écriture qui, pour rendre les romans plus *vivants*, préconise de faire passer les informations au lecteur en les lui faisant vivre plutôt qu'en les lui expliquant.

3. Ces deux exemples, comme tous les exemples de chapitre, sont issus d'un corpus extrait du site AlloCiné par Théophile Blard et diffusé sur la plateforme HuggingFace.

4. Une description de cette expérience a été faite dans un article intitulé « Mining and summarizing customer reviews » publié dans la conférence *Knowledge Discovery and Data mining* en 2004.

5. « Detecting Emotional Contagion in Massive Social Networks », *PlosOne* 2014.

6. Par exemple dans les travaux de Matthew L. Jockers qui sont résumés au lien suivant : <https://www.matthewjockers.net/2015/02/02/syuzhet/>.

7. Cette formule est issue du livre I.A. *La Plus Grande Mutation de l'Histoire* de Kai-Fu Lee qui continue la métaphore : « l'électricité de demain, omniprésente et indispensable ». Kai-Fu Lee qualifie également les données de nouvelle « ressource naturelle ».

8. Le diable se loge dans les détails avec les notations : il ne faut pas confondre w^y qui désigne un vecteur contenant l'ensemble des paramètres de l'étiquette y et w_i^y qui désigne le i -ème de ces paramètres et qui est donc un nombre.

9. Les composantes des mots absents du document seront toutes égales à 0 dans le vecteur représentant celui-ci, ce qui *annulera* le poids correspondant dans le calcul du score (ajouter 0 ne change pas le score !).

10. « *The first machine which is capable of having an original idea* ». Frank Rosenblatt construisit effectivement une machine, le Mark I, qui apprenait à reconnaître la forme des objets (triangulaire, carrée, etc.) présentés devant une caméra.

11. « *Indeed, it strikes us as the first serious rival to the human brain ever devised.* » *The New Yorker*, 6 décembre 1968.

12. La contribution d'une telle caractéristique au score d'une étiquette devra, par exemple, être nulle tant que la caractéristique est inférieure à un certain seuil, et varier ensuite en fonction de la valeur de la caractéristique.

Cette *rupture* ne peut pas être modélisée par une fonction non linéaire, sauf si elle est directement encodée dans les caractéristiques : il est tout à fait possible de définir une composante du vecteur représentant une observation qui vaut 0 si une caractéristique est plus petite qu'un seuil donné, et 1 sinon. Mais ce seuil devra être fixé manuellement et ne saura être déterminé automatiquement à partir des données.

13. En plus de ce résultat, la théorie VC introduit plusieurs notions particulièrement importantes, notamment la notion de *capacité* d'un classifieur qui quantifie le lien entre la taille de l'ensemble d'apprentissage, le nombre de paramètres du classifieur et sa capacité à généraliser.

14. Tous les résultats de la théorie sont des résultats *probabilistes* : contrairement aux théorèmes habituels qui énoncent des résultats qui sont toujours vrais, les théorèmes de la théorie de l'apprentissage ne sont vrais qu'avec « une forte probabilité ». En effet, ceux-ci dépendent toujours du choix de l'ensemble d'apprentissage, et il est possible (même si c'est improbable) que celui-ci ne contienne que des exemples ne permettant pas d'inférer un *bon* classifieur.

15. De manière générale, le gradient d'une fonction est un outil fondamental de l'optimisation. Il décrit les variations d'une fonction autour d'une valeur donnée de ses entrées et permet de déterminer comment modifier celles-ci pour réduire (ou augmenter) la valeur de la fonction. Le gradient de la fonction qui exprime l'erreur en fonction des paramètres du classifieur permet de savoir si une modification d'un des paramètres du classifieur augmentera (ou diminuera) l'erreur faite par le classifieur et servira donc à déterminer la meilleure manière de modifier les paramètres.

16. La question de l'attribution d'auteurs est une question essentielle dans le champ des études littéraires. Elle s'appuie de plus en plus sur des analyses statistiques plus ou moins automatisées, comme l'illustrent les études récurrentes sur la paternité des œuvres de Molière (voir par exemple l'article de Florian Cafiero et Jean-Baptiste Camps « Why Molière most likely did write his plays », *ScienceAdvance*, 2019). Ces approches ont également été appliquées, dans le cadre de la linguistique médico-légale (*forensic linguistics*), à des enquêtes criminelles, par exemple l'affaire du petit Grégory, comme le relate le *Huffington Post* dans un article daté du 16 décembre 2020. Cette utilisation continue toutefois à soulever de nombreuses interrogations et critiques.

17. Les annotations syntaxiques permettent, par exemple, d'identifier le sujet et les compléments d'objets direct ou indirect des différents verbes de la phrase. Les annotations morphologiques permettent, entre autres, d'identifier, pour chaque mot, son nombre (singulier ou pluriel), son genre

(le tchèque distingue 4 genres : féminin, neutre, masculin animé et masculin inanimé) et son cas (parmi les 7 cas possibles).

18. Le *Prague Dependency Treebank* est un projet universitaire porté par des chercheurs de l'Université Charles de Prague qui présente la particularité de ne pouvoir être annoté que par des personnes ayant une bonne connaissance de la langue et de la grammaire de celle-ci. Les couts d'annotation sont donc plus élevés que pour la plupart des corpus nécessaires à l'apprentissage mais c'est l'un des seuls pour lesquels ces couts ont été documentés. À titre de comparaison, le corpus ImageNet contient plus de 14 millions d'images annotées manuellement avec les objets qu'elles contiennent (près de 22 000 catégories d'objets sont définis). Ce travail d'annotation ne nécessitant pas de connaissance particulière a pu être effectué sur la plateforme Mechanical Turk d'Amazon. Il a nécessité le recrutement (sur une période de 3 ans) de plus de 49 000 personnes. Aucune estimation du cout total de cette annotation n'a été publiée.

19. Voir, à ce sujet, le livre d'Antonio A. Casilli, *En attendant les robots* (Seuil, 2019), qui dresse un état des lieux édifiant des conditions de travail des producteurs de données et décrit de manière très documentée la volonté de cacher ce *cyberprolétariat* derrière la promesse d'algorithmes autonomes.

20. « En France, plus de 260 000 « travailleurs du clic » dans le numérique », *Le Figaro*, 22 février 2019.

21. Perrigo B., « Exclusive: OpenAI Used Kenyan Workers on Less Than \$2 Per Hour to Make ChatGPT Less Toxic », *Time*, 18 janvier 2023.

22. « Elon Musk s'en prend à Microsoft et l'accuse d'utilisation abusive des données fournies par Twitter », *Le Monde*, 19 mai 2023. Microsoft aurait téléchargé plus de 26 milliards de tweets en 2022.

23. « La plateforme va limiter la lecture à 10 000 messages par jour pour les comptes vérifiés et 1 000 pour les autres », *Le Figaro*, 1^{er} juillet 2023.

24. OpenSubtitles (www.opensubtitles.org) est un site communautaire permettant de partager des sous-titres et leurs traductions.

25. Les traductions peuvent facilement être extraites à partir des informations qui indiquent quel texte doit être affiché à un moment donné du film (les *timestamps*) : les textes dans deux langues affichées à un même moment sont très certainement des traductions de l'une vers l'autre.

26. Le site Opus (opus.nlpl.eu) recense l'ensemble des corpus parallèles (contenant des phrases et leurs traductions) librement accessibles.

Apprendre à représenter le monde

1. L'opération n'est définie que dans les cas où les dimensions de la matrice et du vecteur sont *compatibles*.

2. Formellement, la fonction ReLU est définie comme : $\text{ReLU}(x) = \max(x, 0)$.

3. Pour détecter si les mots décrits par la i -ème et la j -ème caractéristique sont présents simultanément dans un même texte décrit par un vecteur *one-hot* x , il suffit de calculer $x_i + x_j - 1,5$ en choisissant une colonne de la matrice construisant la représentation cachée de la manière suivante : tous les éléments sont nuls sauf le i -ème et le j -ème, qui sont égaux à 1 ; le biais est fixé à $-1,5$. Les caractéristiques étant binaires, cette combinaison linéaire peut valoir $-1,5$ (si x_i et x_j sont tous les deux nuls), $-0,5$ (si l'un des deux mots est présent) ou $0,5$ si les deux mots sont présents et que, par conséquent, x_i et x_j sont tous les deux égaux à 1. L'application de la fonction ReLU permet alors d'annuler la caractéristique.

4. Cybenko G. « Approximation by superpositions of a sigmoidal function ». *Math. Control Signal Systems* 2, 303–314 (1989). Hornik K., Stinchcombe M., White H. « Multilayer Feedforward Networks are Universal Approximators ». *Neural Networks* 2, 359–366 (1989).

5. Cité par Marcus du Sautoy dans *Le code de la créativité* (Héloïse d'Ormesson, 2020)

6. https://www.lemonde.fr/economie/article/2023/05/30/dans-la-bataille-des-semi-conducteurs-le-succes-de-l-intelligence-artificielle-rebat-les-cartes-de-la-concurrence_6175462_3234.html

7. Il est important de noter que, en 2018, deux des trois récipiendaires du prix travaillaient pour un des GAFAM au moment où le prix Turing leur fut remis alors qu'ils avaient fait l'essentiel de leurs carrières et de leurs découvertes alors qu'ils travaillaient dans des Universités : Yann Le Cun était le *Chief IA Scientist* à Meta (anciennement Facebook) et Geoffrey Hinton était *VP and Engineering Fellow* à Alphababet. Seul Yoshua Bengio travaillait encore dans le monde universitaire (à l'Université de Montréal). Comme nous en parlerons au chapitre 5, Hinton a depuis quitté son poste à Alphabet pour pouvoir exprimer librement ses craintes liées au développement des méthodes d'IA.

8. <https://awards.acm.org/about/2018-turing>

9. Tuan M., « Comment le « deep learning » révolutionne l'intelligence artificielle », *Le Monde*, 24 juillet 2015.

10. ImageNet contient aujourd'hui plus de 14 millions d'images représentant plus de 20 000 objets différents. En 2015, une équipe de Microsoft proposa un réseau de neurones capables d'identifier correctement

les objets dans plus de 95 % des cas, dépassant les performances des humains sur cette tâche. En 2017, la campagne d'évaluation sous la forme que nous décrivons fut arrêtée, la tâche étant considérée comme résolue.

11. <http://host.robots.ox.ac.uk/pascal/VOC/>

12. Pour éviter les problèmes de droit d'auteur, le ImageNET ne contient que les URL des images (plutôt que les images à proprement parler). Chaque personne voulant utiliser le corpus devra télécharger les images. En plus de respecter les droits d'auteur, cette solution facilite également la mise à jour du corpus et évite d'avoir à stocker toutes les images, et donc de limiter les besoins en espace de stockage et en bande passante.

13. Pour être plus précis, word2vec est une famille d'architectures regroupant plusieurs modèles. Nous ne parlons ici que de l'architecture la plus simple, appelée CBOW pour *Continuous Bag of Words*. L'architecture de word2vec la plus couramment employée, *skip-gram*, suit exactement le même principe. La seule différence réside dans la tâche pour laquelle elle est utilisée : plutôt que de chercher à prédire le mot au centre de la suite de mots, son objectif est de prédire les 4 mots du contexte, connaissant le mot au centre.

14. Un robot d'indexation (*web spider*) est un programme qui explore automatiquement l'Internet (essentiellement en collectant et suivant les hyperliens dans les documents) en téléchargeant les pages visitées. Ces robots sont une brique essentielle de tous les moteurs de recherche : ce sont eux qui vont construire l'index, c'est-à-dire la base de données contenant toutes les pages connues par le moteur de recherche, et qui pourront éventuellement être retournées lors de la requête d'un utilisateur.

15. Cette version est toujours accessible à l'url <https://code.google.com/archive/p/word2vec/>

16. <https://en.wikipedia.org/wiki/Special:Statistics>

17. La projection d'un vecteur est une opération mathématique permettant de réduire sa dimension. Des méthodes de projection adaptées aux plongements lexicaux ont été développées afin de pouvoir représenter ceux-ci en 2 dimensions (et donc pouvoir être visualisés facilement) tout en conservant la géométrie de l'espace de départ : des mots avec des plongements lexicaux proches auront alors des projections proches.

18. Tainturier B., « Islamo-gauchisme : carrière médiatique d'une notion polémique », *La Revue des médias*, 27 novembre 2020.

19. Alexandre Gefen explique en détail comment il a utilisé word2vec (puis ChatGPT) pour définir le romantisme dans son ouvrage *Vivre avec ChatGPT* (L'Observatoire, 2023).

20. <https://www.cairn.info/revue-document-numerique-2017-2-page-31.htm>. Cette étude ne porte que sur les articles publiés dans des conférences ou des revues de TAL. Les articles de Tomáš Mikolov et ses co-auteurs ont été publiés dans des conférences d'apprentissage statistique et n'y sont donc pas comptabilisés.

21. Bahdanau D., Cho K. et Bengio Y., « Neural Machine Translation by Jointly Learning to Align and Translate », *arXiv*: 1409.0473.

22. Vaswani A. *et al.*, « Attention is all you need », *arXiv*: 1706.03762. L'article a été cité, au moment où ces lignes sont écrites, plus de 90 000 fois, devenant l'un des articles les plus cités en IA, voire en informatique.

23. Des chercheurs ont par exemple utilisé des transformeurs pour construire des programmes qui génèrent automatiquement les noms de molécules à partir de leurs formules (Krasnov L., Khokhlov I., Fedorov M.V. *et al.*, « Transformer-based artificial neural networks for the conversion between chemical notations », *Sci Rep* 11, 14798, 2021) ou pour prédire les réarrangement d'atomes lors des réactions chimiques (Schwaller P. *et al.*, « Extraction of organic chemistry grammar from unsupervised learning of chemical reactions », *Sci. Adv.* 7, 2021).

24. Rogers A., Kovaleva O. et Rumshisky A. « A Primer in BERTology: What We Know About How BERT Works », *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

Les surprenants pouvoirs des transformeurs

1. Il n'y a, en pratique, qu'une modification à apporter à l'algorithme d'apprentissage : il faut s'assurer que celui-ci voit autant de phrases de chaque langue. Les exemples ne sont donc pas complètement choisis au hasard (auquel cas le modèle verrait beaucoup plus de phrases en anglais, très largement majoritaires dans les corpus considérés aujourd'hui), mais de manière à garantir que les langues soient également représentées.

2. Codex, le modèle de code d'OpenAI, est ainsi appris sur 159 Go de Python, Code-Llama, le modèle de Meta, sur 938 Go de code et 3,5 To de textes.

3. <https://openai.com/research/dall-e>

4. Il faut bien garder à l'esprit que tous ces développements se sont faits de manière concomitante : DALL·E repose en partie sur GPT-3 pour modéliser le langage, l'un des plus grands modèles de langue existant actuellement.

5. Reddit est un service en ligne qui permet aux utilisateurs de partager du contenu, de participer à des discussions et de créer des communautés en

ligne autour d'une variété de sujets. Les participants peuvent voter sur le contenu, ce qui contribue à le classer en fonction de sa pertinence. En septembre 2021, Reddit regroupait plus de 430 millions d'utilisateurs actifs mensuels, et plus de 10 milliards de messages avaient été postés.

6. Les capacités de GPT-2 et l'émerveillement des journalistes devant la qualité des textes générés peuvent, entre autres, se voir dans un article de *The Verge* (Vincent J., « OpenAI's new multitasking AI writes, translates, and slanders », 18 décembre 2020) ou du *Guardian* (Hern A., New AI fake text generator may be too dangerous to release, say creators, 14 février 2019)

7. Tual M., « Fausses informations : un programme d'IA jugé trop dangereux pour être rendu public », *Le Monde*, 19 février 2019.

8. Au moment où ces lignes sont écrites, aucune des informations mentionnées dans la suite n'a été confirmée officiellement par OpenAI.

9. Pour être plus précis, tous les modèles sont entraînés avec les mêmes tâches et données, la différence réside uniquement dans l'ordre et la fréquence avec lesquels ces tâches sont considérées.

10. <https://arxiv.org/abs/2101.03961>. Il est intéressant de voir que deux des auteurs de l'article (William Fedus et Trevor Cai) ont été débauchés de Google et travaillaient à OpenAI au moment de la conception de GPT-4.

11. Vilar D. *et al.*, « Prompting PaLM for Translation: Assessing Strategies and Performance », <https://aclanthology.org/2023.acl-long.859.pdf>.

12. <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>

13. Perrigo B., « Exclusive: OpenAI Used Kenyan Workers on Less Than \$2 Per Hour to Make ChatGPT Less Toxic », *Time*, 18 janvier 2023.

14. Douet M., « Au Kenya, des « entraîneurs » de ChatGPT s'élèvent contre leurs conditions de travail », *Le Monde*, 19 octobre 2023.

ChatGPT et les giga-modèles, une révolution ou une évolution ?

1. « Intelligence artificielle dans les médias : l'effet ChatGPT », *Aday*, 14 juin 2023.

2. Alcaraz M., « OpenAI, derrière ChatGPT, valorisé 90 milliards de dollars », *Les Echos*, 27 septembre 2023.

3. « Google teste un outil d'intelligence artificielle pour aider les journalistes à écrire des articles de presse », *Le Monde* avec *AFP*, 21 juillet 2023. Des journaux français vont également expérimenter cette utilisation de

ChatGPT : Demagny X., Baldacchino J., « L'Est républicain suspend son projet d'expérimenter ChatGPT pour relire les articles des correspondants locaux », *France Inter*, 26 octobre 2023.

4. Talbot C., « Aux États-Unis, les « chatbots » désormais intégrés dans la vie des entreprises : « C'est un nouveau membre de l'équipe de direction » », *Le Monde*, 7 juin 2023.

5. Mesmer P., « Le manga « Black Jack » ressuscite grâce à l'intelligence artificielle », *Le Monde*, 4 août 2023.

6. Faure G., « ChatGPT, qu'est-ce qu'on fait à dîner ce soir ? », *Le Monde*, 7 octobre 2023.

7. « Lyon : ChatGPT utilisé par la moitié des élèves de Master d'un professeur pour rédiger leurs devoirs ». <https://intelligence-artificielle.developpez.com/actu/340274/Lyon-ChatGPT-utilise-par-la-moitie-des-eleves-de-Master-d-un-professeur-pour-rediger-leurs-devoirs-Cette-pratique-m-inquiete-Elle-pose-un-veritable-probleme-d-integration-des-savoirs/>

8. « ChatGPT réussit l'entretien de codage Google pour un ingénieur de niveau 3 avec un salaire de 183 000 \$ ». <https://intelligence-artificielle.developpez.com/actu/341183/ChatGPT-reussit-l-entretien-de-codage-Google-pour-un-ingenieur-de-niveau-3-avec-un-salaire-de-183-000-tout-en-indiquant-qu-il-ne-peut-pas-reproduire-la-creativite-humaine/>

9. Larousserie D., « Les défauts de ChatGPT sous la loupe des scientifiques », *Le Monde*, 18 octobre 2023.

10. Voir par exemple : Rocca N., « Des données sensibles de Samsung divulgués sur ChatGPT par des employés », *RFI*, 9 avril 2023, ou « Samsung interdit l'utilisation de ChatGPT à une partie de ses employés », *Le Monde avec AFP*, 2 mai 2023.

11. Zuboff S., *L'âge du capitalisme de surveillance*, Zulma, 2020.

12. <https://authorsguild.org/news/ag-and-authors-file-class-action-suit-against-openai/>

13. Piquart A., « IA : Microsoft s'engage à couvrir ses clients en cas de procès pour droits d'auteur », *Le Monde*, 7 septembre 2023.

14. Piquard A., « Intelligence artificielle : des médias bloquent l'accès de leurs sites aux robots de ChatGPT et d'OpenAI », *Le Monde*, 30 août 2023.

15. Freyssenet E. et Alcaraz M., « Intelligence artificielle : la presse cherche une alternative au pillage », *Les Echos*, 11 janvier 2024.

16. Dèbes F., « Intelligence artificielle : la France se lance dans la bataille culturelle des données », *Les Echos*, 12 décembre 2023.

- [17.](#) « Google coupe les ponts avec Huawei : ce que ça change pour les utilisateurs », *Le Monde*, 22 mai 2019.
- [18.](#) Bender E. M. *et al.*, « On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? », FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021.
- [19.](#) Perrigo B., « Why Timnit Gebru Isn't Waiting for Big Tech to Fix AI's Problems », *Time*, 18 janvier 2022.
- [20.](#) Luccioni A., Viguier S. et Liguozat A.-L., « Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model », *arXiv*, 3 novembre 2022.
- [21.](#) Pattel D. et Ahmad A., « The Inference Cost Of Search Disruption – Large Language Model Cost Analysis », *semianalysis*, 9 février 2023.
- [22.](#) Larousserie D., « Les concurrents de ChatGPT se multiplient », *Le Monde*, 6 juin 2023.
- [23.](#) Bender E. M. et Koller A., « Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data ». In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- [24.](#) Voir [note 16](#) du chapitre 5.
- [25.](#) Voir, par exemple, Don, A., Brennan, S., Laurel, B., & Shneiderman, B. (1992, June). Anthropomorphism: from ELIZA to Terminator 2. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 67–70)
- [26.](#) « ChatGPT generates 'convincing' fake scientific article », *JMIR Publications*, 5 juillet 2023.
- [27.](#) <https://x.com/sama/status/1601731295792414720?s=20>
- [28.](#) <https://arxiv.org/abs/2210.13382>
- [29.](#) Desnoyers F., « L'intelligence artificielle, cet “ogre numérique” qui angoisse les travailleurs », *Le Monde*, 8 novembre 2023.
- [30.](#) Boutelet C., « Le quotidien allemand *Bild* réduit ses coûts et mise sur l'intelligence artificielle », *Le Monde*, 28 juin 2023.
- [31.](#) Akst D., « Automation anxiety », *The Wilson Quarterly* (1976-), vol. 37, no. 3, 2013. JSTOR, <https://www.jstor.org/stable/wilsonq.37.3.06>.
- [32.](#) David H., « Why Are There Still So Many Jobs? The History and Future of Workplace Automation », *The Journal of Economic Perspectives*, vol. 29, no. 3, 2015, pp. 3–30. JSTOR, <http://www.jstor.org/stable/43550118>.
- [33.](#) Voici, par exemple, les consignes d'écriture que doivent respecter tous les documents rédigés par la Commission Européenne :

https://commission.europa.eu/resources-partners/europa-web-guide/design-content-and-development/content/web-writing-guidelines_en

34. Dell'Acqua F. *et al.*, « Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality », *Harvard Business School Technology & Operations Mgt. Unit Working Paper*, 2015, n° 24-013.

35. Hui X., Reshef O. et Zhou L., « The Short-Term Effects of Generative Artificial Intelligence on Employment: Evidence from an Online Labor Market », *SSRN*, 31 juillet 2023.

36. Les acteurs étaient plus inquiets de l'utilisation récente des méthodes d'IA permettant de créer de véritables avatars virtuels, qui ont été, par exemple, utilisés pour faire rejouer des acteurs décédés depuis plusieurs années (comme Peter Cushing dans *Rogue One*) et qui laisse penser qu'il sera bientôt envisageable de faire des films sans acteur.

37. « Geoffrey Hinton, pionnier de l'IA, quitte Google et dit « regretter » son invention », *Le Monde avec AFP*, 2 mai 2023.

38. Eytan, « Ce qu'Elon Musk pense réellement de l'intelligence artificielle », *Forbes*, 2 février 2022.

39. <https://www.safe.ai/statement-on-ai-risk>

40. Bostrom N., *Superintelligence*, Dunod, 2017.

41. Noiry N., « Des biais de représentativité en intelligence artificielle », *Binaire (Le Monde)*, 31 août 2021.

42. Ceux-ci couvrent quatre types d'applications jugées contraires aux « valeurs » européennes : les systèmes d'identification biométrique à distance, les systèmes de crédit social, l'exploitation des vulnérabilités liées à l'âge ou au handicap physique ou mental, et les techniques subliminales pour altérer un comportement.

43. Alcaraz M et Dèbes F., « AI Act : le droit d'auteur, pomme de discorde du compromis européen », *Les Echos*, 16 décembre 2023.

Épilogue

1. Browne R., « 'Overhyped' generative AI will get a 'cold shower' in 2024, analysts predict », *CNBC*, 10 octobre 2023.

INDEX

accord sujet-verbe [1](#)
AfriBERT [1](#)
algorithme
 d'apprentissage [1](#) [2](#)
 du perceptron [1](#) [2](#) [3](#) [4](#) [5](#)
AlphaGo [1](#) [2](#)
Altman, Sam [1](#)
Amazon Mechanical Turk [1](#) [2](#)
amorce [1](#) [2](#)
analyse
 de polarité [1](#)
 de sentiments [1](#) [2](#) [3](#) [4](#) [5](#) [6](#)
 discriminante linéaire [1](#)
 d'opinion [1](#)
annotation des données [1](#)
apprentissage
 auto-supervisé [1](#) [2](#) [3](#)
 automatique [1](#) [2](#)
 bout-à-bout [1](#)
 par renforcement [1](#)
 profond (*deep learning*) [1](#) [2](#) [3](#) [4](#)

statistique [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#)
[15](#) [16](#) [17](#) [18](#) [19](#) [20](#)

symbolique [1](#)

apprentissage supervisé [1](#)

approximateur universel [1](#)

arbre de dépendances [1](#) [2](#)

architecture à propagation avant [1](#)

ARPA [1](#) [2](#)

attention croisée [1](#) [2](#)

auto-supervision [1](#)

Baseball (IA) [1](#) [2](#) [3](#) [4](#) [5](#)

Bender, Emily [1](#) [2](#) [3](#)

Bengio, Yoshua [1](#) [2](#) [3](#) [4](#)

BERT [1](#) [2](#) [3](#) [4](#) [5](#)

BERTeus [1](#)

BERTje [1](#)

Biden, Joe [1](#)

Bing [1](#)

biotraduction [1](#)

BLOOM [1](#)

Bostrom, Nick [1](#)

ByteDance [1](#)

CamemBERT [1](#)

Casilli, Antonio [1](#) [2](#)

chaînes de traitement [1](#) [2](#) [3](#)

chatbots [1](#)

ChatGPT [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#)

Chavalarias, David [1](#)

Chervonenkis, Alexey [1](#)
Chinchilla [1](#)
Chomsky, Noam [1](#)
classifieur linéaire [1](#) [2](#) [3](#) [4](#) [5](#)
CLIPS [1](#)
cloud computing [1](#)
Collobert, Ronan [1](#)
compositionnalité [1](#)
conférence de Dartmouth [1](#)
Copilot [1](#)
couche
 à propagation avant [1](#) [2](#)
 d'auto-attention [1](#) [2](#)
 d'encodage [1](#) [2](#) [3](#) [4](#)
DALL·E [1](#) [2](#) [3](#) [4](#) [5](#)
Databricks [1](#)
data journalism [1](#)
datamining [1](#)
Davos, forum [1](#)
déclinaisons [1](#)
décodeur [1](#)
deep fake [1](#)
DeepL [1](#)
deep learning Voir apprentissage profond [1](#)
DeepMind [1](#)
de Saussure, Ferdinand [1](#)
descente de gradient stochastique [1](#)
désinformation [1](#)

diffusion stable [1](#)
digital humanities [1](#)
distance euclidienne [1](#)
distribution des mots [1](#)
DuckDuckGo [1](#)
Eliza [1](#) [2](#) [3](#) [4](#) [5](#) [6](#)
encodage [1](#)
encodeur [1](#)
Facebook [1](#)
feed-forward [1](#)
few-shot learning [1](#)
FinBERT [1](#)
Firth, John [1](#)
Fisher, Ronald [1](#)
FlauBERT [1](#)
GAFAM [1](#) [2](#)
Gebru, Timnit [1](#)
Gefen, Alexandre [1](#)
giga-modèles [1](#) [2](#) [3](#) [4](#) [5](#) [6](#)
 toute-puissance [1](#)
GitHub [1](#)
Google [1](#) [2](#) [3](#) [4](#)
Google Brain [1](#)
Google Images [1](#)
Google Research [1](#)
Google Translate [1](#) [2](#)
GPT-1 [1](#)
GPT-2 [1](#) [2](#)

GPT-3 [1](#) [2](#) [3](#)
GPT-4 [1](#) [2](#) [3](#)
guerre froide [1](#)
Harris, Zellig [1](#)
Hassabis, Demis [1](#) [2](#)
Hinton, Geoffrey [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#)
hiver de l'IA [1](#)
Huawei [1](#)
Hugging Face [1](#)
humanités numériques [1](#)
hyper-paramètres [1](#)
IA symbolique [1](#)
Imagen [1](#)
ImageNet [1](#) [2](#) [3](#) [4](#) [5](#)
ingénierie des caractéristiques [1](#)
InstructGPT [1](#)
instructions fine-tuning [1](#)
interactions homme-machine [1](#)
inverse document frequency [1](#)
Joos, Martin [1](#)
journalisme de données [1](#)
Kim, Najoung [1](#)
Koller, Alexander [1](#)
langue agglutinante [1](#)
Le Cun, Yann [1](#) [2](#) [3](#)
Le Ludec, Clément [1](#)
Li, Fei-Fei [1](#)
linguistique médico-légale [1](#)

Linzen, Tal [1](#)
LlaMA [1](#), [2](#)
LoRA [1](#)
Lunar (programme) [1](#), [2](#), [3](#), [4](#), [5](#)
machine learning [1](#)
matrice [1](#), [2](#)
 de score [1](#)
McCarthy, John [1](#)
McCulloch, Warren [1](#)
McMillan-Major, Angelina [1](#)
mécanisme d'attention [1](#)
 croisée [1](#)
mécanisme d'auto-attention [1](#)
Microsoft [1](#), [2](#), [3](#), [4](#), [5](#)
Microworkers [1](#)
MidJourney [1](#)
Mikolov, Tomáš [1](#), [2](#), [3](#)
minimisation du risque empirique (ERM) [1](#), [2](#), [3](#), [4](#)
Mistral AI [1](#)
Mitchell, Margaret [1](#)
modèle de langue [1](#), [2](#), [3](#), [4](#)
modèle pré-entraîné [1](#)
modèles GPT [1](#)
modèles multilingues [1](#), [2](#), [3](#), [4](#), [5](#)
modèles multimodaux [1](#)
modèle vectoriel [1](#)
mots-outils [1](#)
Musk, Elon [1](#), [2](#)

negative sampling [1](#)
Ney, Herman [1](#)
NSA [1](#)
Office [1](#)
OpenAI [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#),
[18](#)
OpenSubtitles [1](#)
PaLM [1](#)
Pascal VOC [1](#)
perceptron [1](#), [2](#), [3](#), [4](#), [5](#)
perroquets stochastiques [1](#), [2](#)
phase de propagation avant [1](#)
phase de rétropropagation [1](#)
Pichai, Sundar [1](#)
Pitts, Walter [1](#)
plongements lexicaux [1](#), [2](#), [3](#), [4](#)
problème d'optimisation [1](#)
prompt engineering [1](#)
reconnaissance de la parole [1](#)
Reddit [1](#)
ReLU (fonction) [1](#)
renforcement humain [1](#)
représentation cachée [1](#), [2](#), [3](#), [4](#)
réseau à propagation avant [1](#), [2](#)
réseaux de neurones [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#),
[14](#), [15](#), [16](#), [17](#), [18](#), [19](#), [20](#), [21](#), [22](#), [23](#), [24](#), [25](#), [26](#), [27](#), [28](#), [29](#), [30](#),
[31](#), [32](#), [33](#), [34](#), [35](#), [36](#)
réseaux récurrents [1](#), [2](#)

rétropropagation du gradient [1](#) [2](#) [3](#)
risque empirique [1](#)
robot d'indexation [1](#)
Rosenblatt, Frank [1](#) [2](#)
Shrdlu (programme) [1](#) [2](#) [3](#) [4](#)
Simon, Herbert [1](#)
SpaceX [1](#)
StackOverflow [1](#)
stylométrie [1](#)
Sunak, Rishi [1](#)
systèmes experts [1](#)
term frequency [1](#)
tête de classification [1](#) [2](#) [3](#)
têtes d'attention [1](#)
théorie VC [1](#) [2](#)
TikTok [1](#)
tokenisation [1](#)
tokens [1](#)
traduction automatique [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#)
transformeur [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#)
 génératif pré-entraîné [1](#)
travailleurs du clic [1](#)
Tubaro, Paola [1](#)
Twitter [1](#) [2](#)
unités sous-lexicales [1](#)
Vapnik, Vladimir [1](#)
vecteur [1](#)
 de scores [1](#)

VisionTransformer [1](#)
Weizenbaum, Joseph [1](#) [2](#) [3](#)
Weston, Jason [1](#)
Wikipédia [1](#)
Windows [1](#)
Winograd, Terry [1](#)
Wirk [1](#)
word2vec [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#)
[18](#) [19](#) [20](#) [21](#) [22](#) [23](#)
zero-shot learning [1](#)

Table des Matières

Page de Copyright	2
Table des matières	3
Prologue	5
Chapitre 1. Des programmes pour parler, un défi insurmontable	13
Décrire le langage avec des règles	13
Des règles pour traduire	19
Parler, une tâche si simple et pourtant si compliquée	21
Construire automatiquement le sens des phrases	24
Compter pour comprendre	31
Chapitre 2. Apprendre en faisant : la révolution de l'apprentissage statistique	42
« À voir » ou « à jeter » : déterminer la polarité d'un texte	45
Une théorie pour l'apprentissage statistique	53
Limites des classifieurs linéaires	58
Chapitre 3. Apprendre à représenter le monde	67
Apprendre à représenter : les réseaux de neurones	69
Un réseau de neurones pour représenter les mots : word2vec	80
La révolution des transformeurs	93
Analyser les représentations cachées	107
Chapitre 4. Les surprenants pouvoirs des transformeurs	111
Générer du texte	113
Représenter toutes les langues	119
Des modèles multimodaux	122
L'efficacité déraisonnable des données	127
Un nouveau paradigme pour le TAL	135
Après GPT-3 ? La route vers ChatGPT	138
Et ChatGPT dans tout ça ?	142
Chapitre 5. ChatGPT et les giga-modèles, une révolution ou une évolution ?	144

Une seule limite : l'imagination	144
Les limites propres à ChatGPT	148
Les limites de l'apprentissage auto-supervisé	158
Vivre avec ChatGPT	165
Épilogue	181
Notes et références	184
Index	197