IA Générative et Traitement Automatique des Langues

Enjeux, risques et opportunités pour l'éducation

Claire Gardent

CNRS / LORIA, Nancy

Plan

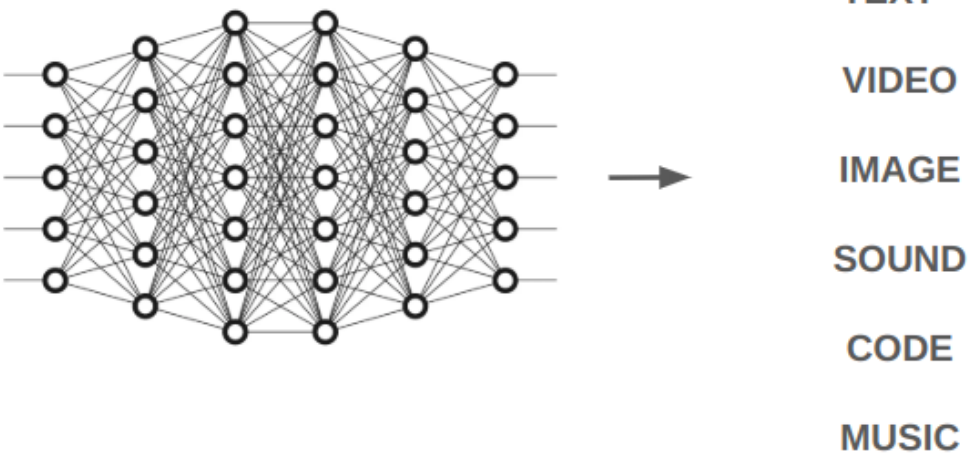
- Qu'est ce que l'IA générative (IAG) ?
- IAG et traitement automatique des langues (TAL)
- TAL et éducation
- Interagir avec les IAG en Python

Generative AI

Qu'est-ce que l'IA générative ?

Une branche de l'IA qui ***génère*** du contenu à l'aide de techniques d'apprentissage automatique :

- **Apprentissage profond** : Utilise des réseaux de neurones
- **Apprentissage (auto)-supervisé** : Nécessite des données d'entraînement (exemples d'entrée/sortie)

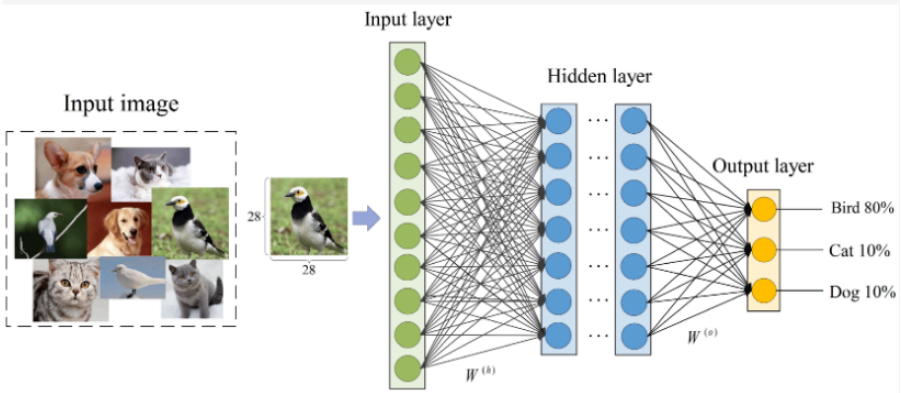


Qu'est-ce qu'un réseau de neurones ?

Réseaux de neurones, apprentissage profond

- Les neurones sont interconnectés dans d'immenses réseaux
- Chaque neurone effectue une tâche simple de reconnaissance de motifs
- Lorsqu'il est activé, le neurone envoie un signal à ses connexions
- La sortie du réseau est déterminée par les neurones activés

Qu'est-ce qu'un réseau de neurones ?



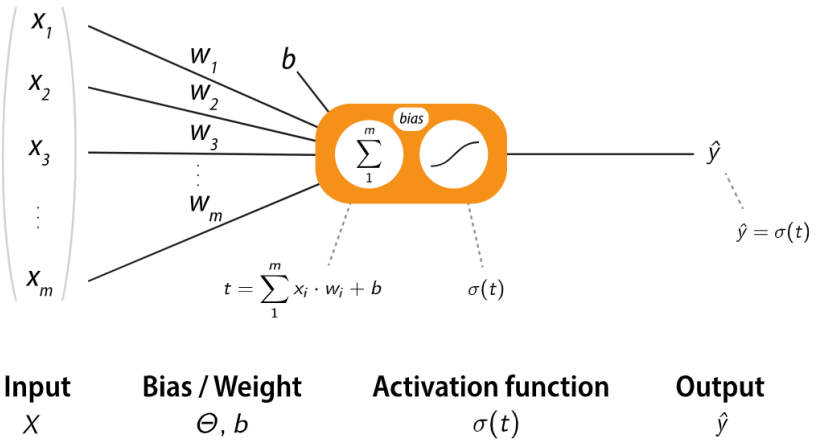
Source : Deng et al. 2022

Un réseau de neurones comporte plusieurs couches :

- **Une couche d'entrée** qui modélise les données d'entrée. Par exemple, pour un classificateur d'images, les pixels de l'image
- **Une couche de sortie** qui représente la prédiction du modèle
- La couche de sortie est souvent une distribution de probabilité. Par exemple, les trois neurones de sortie indiquent la probabilité de chaque classe cible (Oiseau, Chat ou Chien)
- **Une ou plusieurs couches intermédiaires** qui modélisent la relation entre l'entrée et la sortie

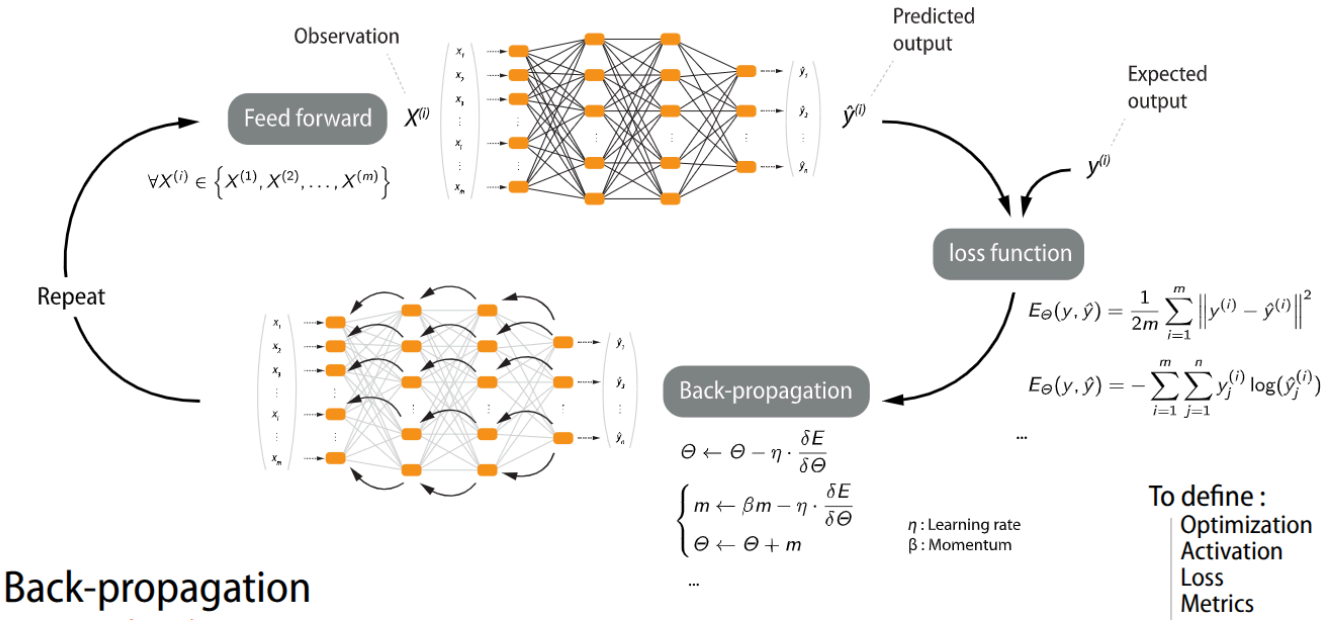
Comment les neurones calculent-ils des valeurs ?

$$\hat{y} = \sigma(\Theta^T \cdot X + b)$$



- Chaque neurone applique une ***fonction d'activation*** à la ***somme pondérée de ses entrées*** pour produire une ***valeur d'activation***.
- Cette ***valeur d'activation*** est transmise comme entrée (signal) à la couche suivante.
- Les ***poids*** sont appris au cours de l'entraînement.

Entraînement - L'algorithme de rétropropagation



Src: Fiddle Course on Deep Learning

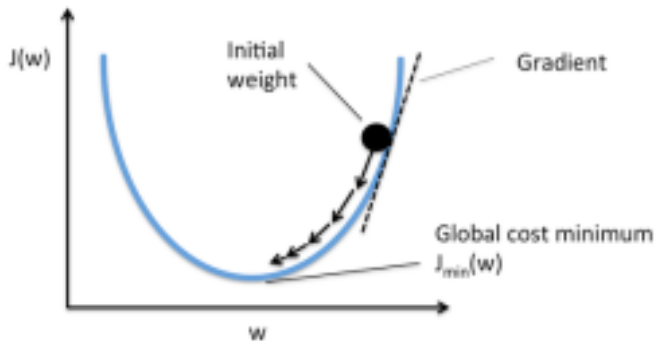
SGD

Descente de Gradient Stochastique (**SGD**)

- Met à jour les poids selon la règle suivante (η = hyperparamètre du taux d'apprentissage) :

$$w \leftarrow w - \eta \frac{dJ(w)}{dw}$$

- Ajuste chaque poids dans la direction de la **dérivée** (***gradient***).



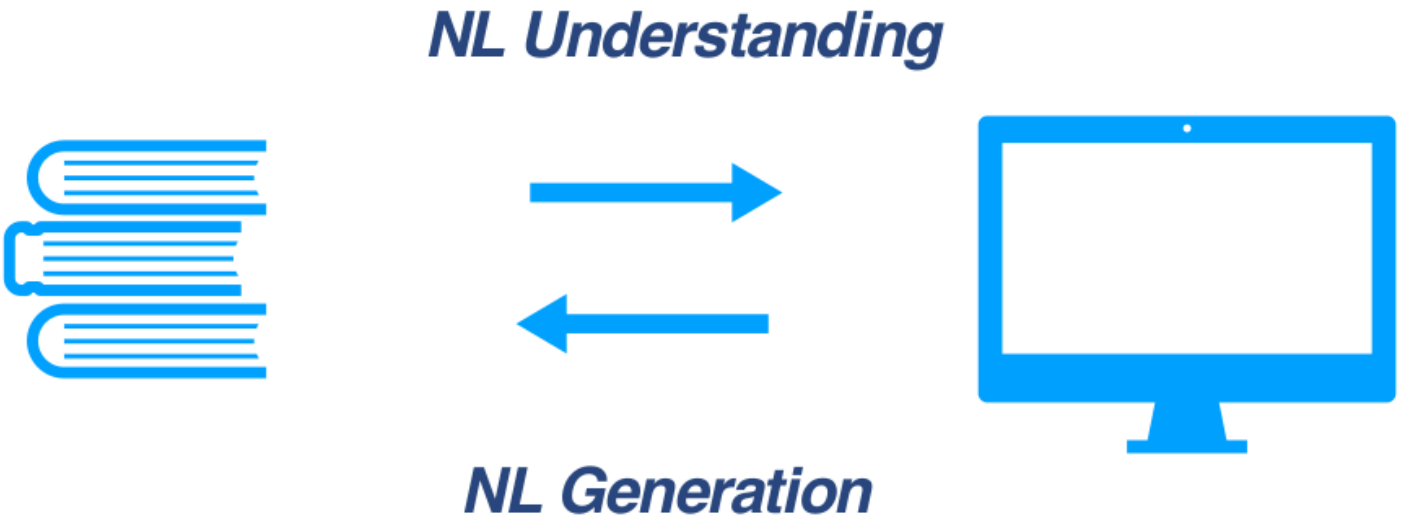
$dJ(w)$ est positif, donc la règle de mise à jour réduit la valeur de w, entraînant une diminution de $J(w)$.

Rumelhart et al. 1986

IA Générative et Traitement Automatique des Langues

Qu'est ce que le TAL ?

Le **Traitement Automatique des Langues (TAL)** est un domaine de l'**Intelligence Artificielle (IA)** qui vise à permettre aux ordinateurs de ***comprendre*** et de ***générer*** le langage humain. Il est également utilisé pour analyser et étudier la structure et l'usage du langage naturel.



Exemples de tâches et applications en TLN

NLU (Compréhension)

- Détection de spam (Filtrage des emails)
- Analyse des sentiments (Surveillance des réseaux sociaux)
- Classification de texte
- Attribution d'auteur
- Extraction d'information
- Moteurs de recherche (Google, Bing)

L'entrée est un texte

NLG (Génération)

- Dialogue Humain-Machine (Support client automatisé, Chatbots)
- Traduction (DeepL, Google Translate)
- Simplification de texte (pour les non-experts, les non-natifs, les personnes ayant des difficultés de lecture)
- Résumé
- Légendes d'images
- Sous-titres vidéo
- Écriture créative (poèmes, romans, essais)

L'entrée est variée : texte, données, données numériques, images, vidéo, etc.

Quatre étapes clés du TAL neuronal

2014 - **Encodeur-Décodeur pour la traduction automatique**

2015 - **Attention croisée**

2017 - **Transformer**

- Pré-entrainement et ajustement fin
- Le parallélisme permet d'entrainer de grands modèles sur des ensembles de données plus volumineux

2023 - **LLMs** (ChatGPT, BLOOM, Llama, LeChat, ...)

2014 - L'architecture encodeur-décodeur

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its

2014 - L'architecture encodeur-décodeur

Encodeur

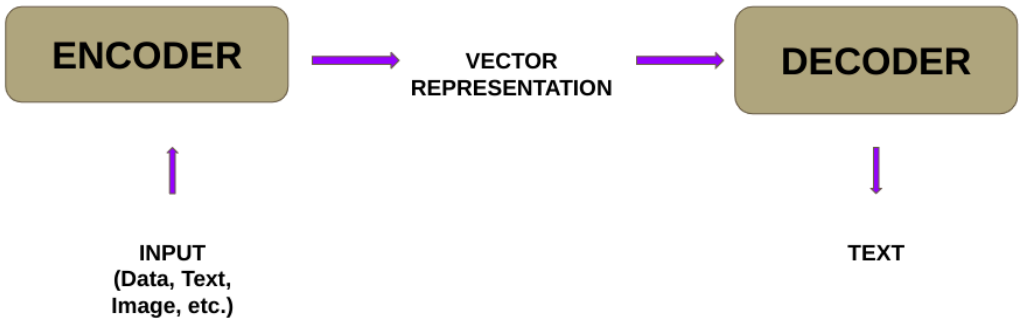
- ***Construit une représentation continue*** de l'entrée, un vecteur à valeurs réelles
- Décodeurs couramment utilisés
  - Récurrents : RNN, LSTM, GRU
  - Convolutionnel
  - Graphique
  - Transformer

Décodeur

- = ***Modèle de langage***
- ***Génère du texte un mot à la fois***
- Conditionné par l'entrée
- Encodeurs couramment utilisés
  - Récurrents : RNN, LSTM, GRU
  - Transformer

L'architecture Encodeur-Décodeur

Un cadre unificateur pour toutes les tâches de génération de texte

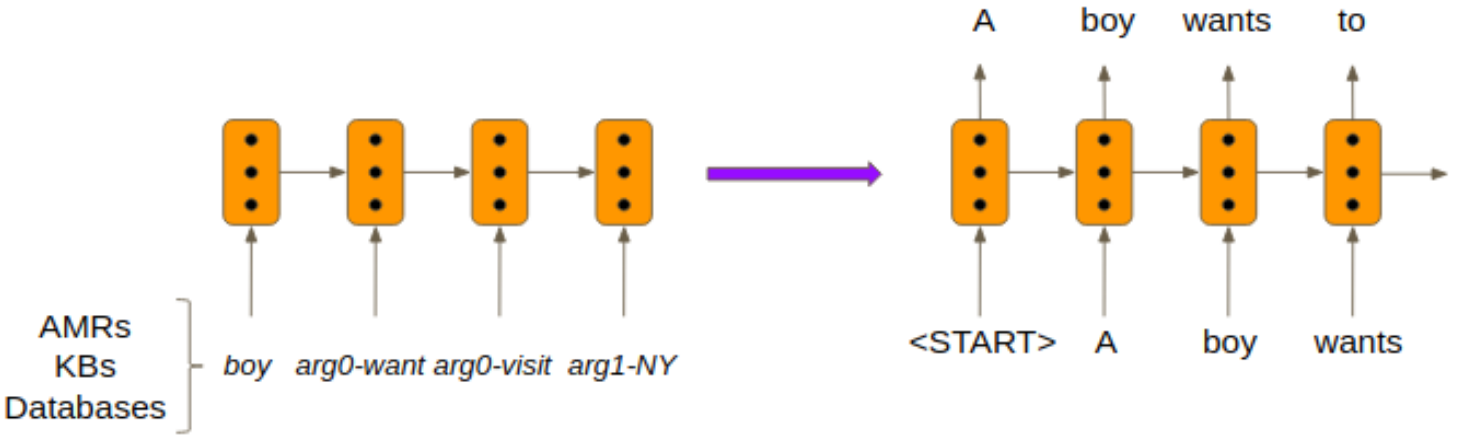


Cadre unificateur pour la génération de texte

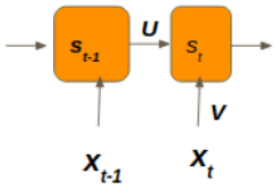
- Les différents types d'entrée (données numériques, texte, graphe, image) sont encodés en une représentation numérique
- Les différentes taches de génération sont traitées par une architecture ***bout en bout*** (Mapping direct de l'entrée à la sortie)

Encodeur récurrent

- L'entrée pour la génération de langage naturel (TALN) (texte, mais aussi données et représentations de signification) est une séquence de tokens
- Les données ou les représentations de signification doivent d'abord être linéarisées



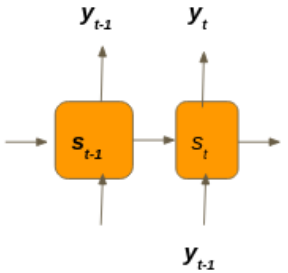
Encodage de l'entrée avec un RNN



$$s_t = tanh(U * s_{t-1} + V * x_t)$$

- x_i sont des vecteurs représentant les tokens d'entrée (mots, données ou tokens de représentation du sens)
- À chaque étape, l'encodeur produit un nouveau vecteur s_t (state) qui représente le contenu de la séquence précédente de tokens
- Le dernier état représente le sens de l'ensemble de l'entrée
- U et V sont les paramètres appris pendant l'entraînement
- tanh est une fonction non linéaire

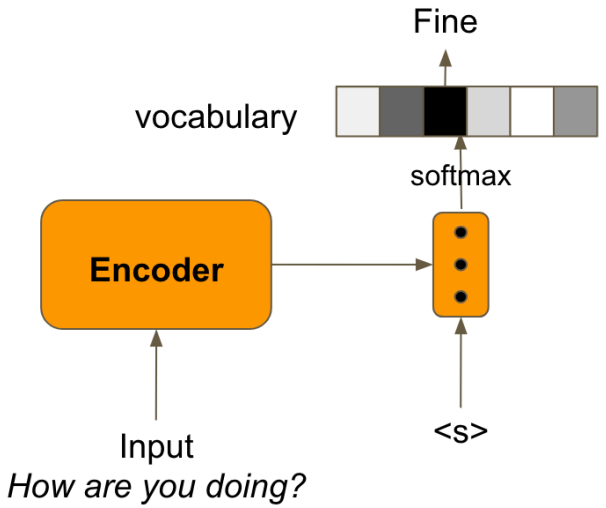
Décodage de mots avec un RNN



$$
\begin{aligned}
s_t &= tanh(U * s_{t-1} + V * y_{t-1}) \\
y_t &= softmax(W * s_{t-1})
\end{aligned}
$$

- y_t est le mot prédit à l'instant t
- s_t est l'état du réseau à l'instant t
- Chaque nouvel état est calculé en tenant compte de l'état précédent $s{t-1} $et du dernier mot prédit $y{t-1}$
- La fonction softmax transforme un vecteur de scores en une distribution de probabilité
- À chaque instant t, le token de sortie/prédit y_t est échantillonné à partir de cette distribution de probabilité

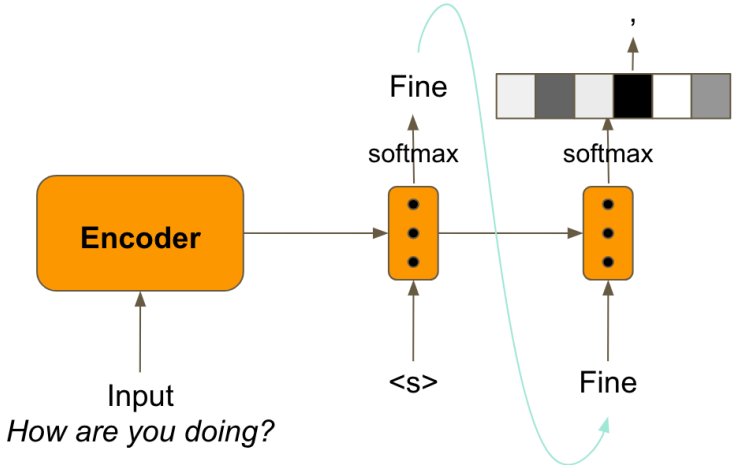
Générer du texte avec un RNN



$$p(\text{Fine}|\text{<s>}, \text{How are you doing?})$$

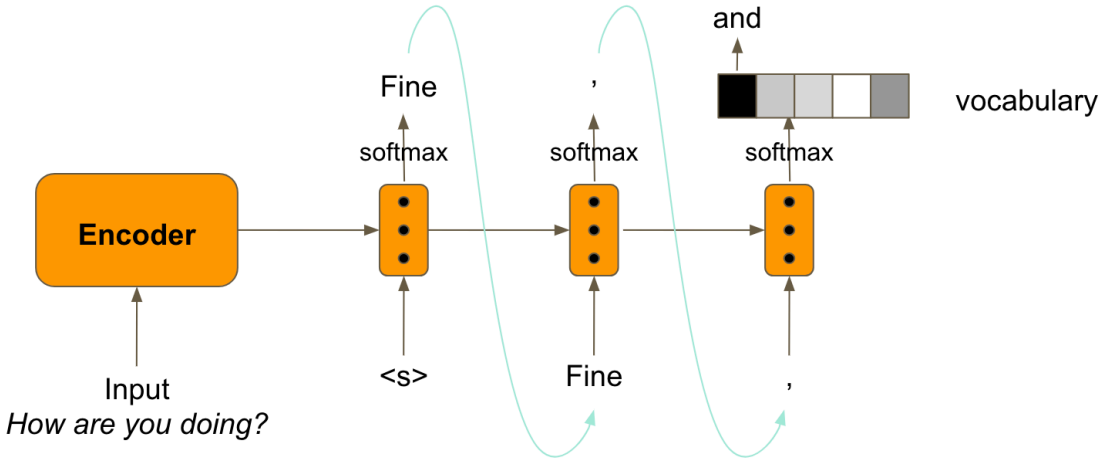
Conditional Generation

Générer du texte avec un RNN



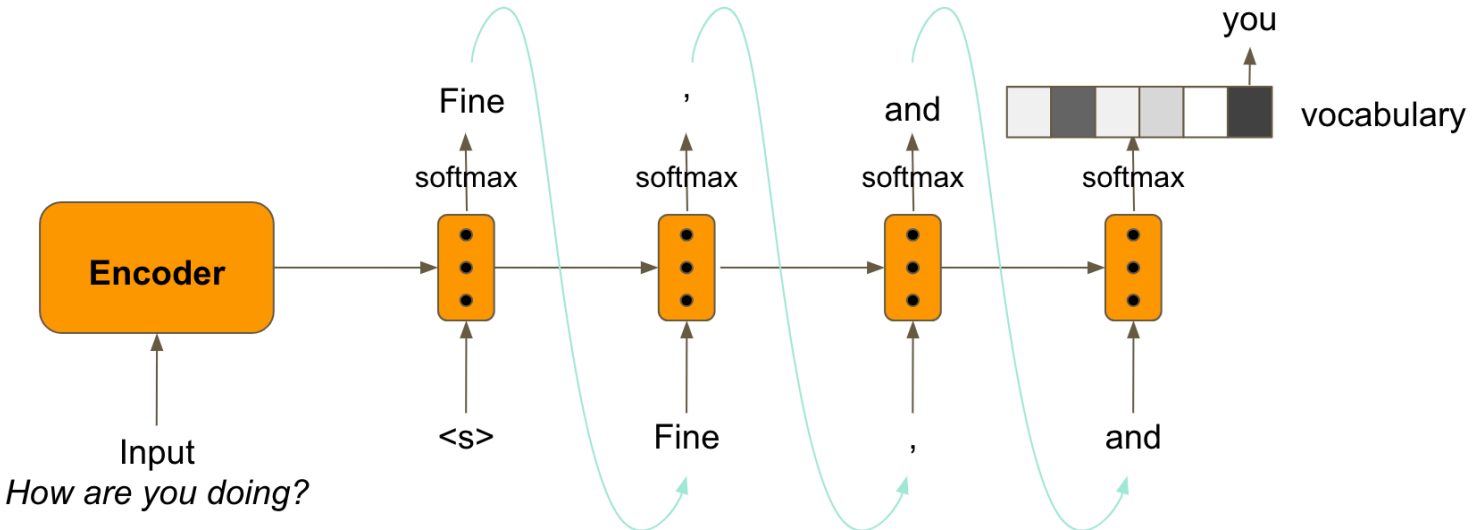
$$p(,|\text{<s> Fine, How are you doing?})$$

Générer du texte avec un RNN

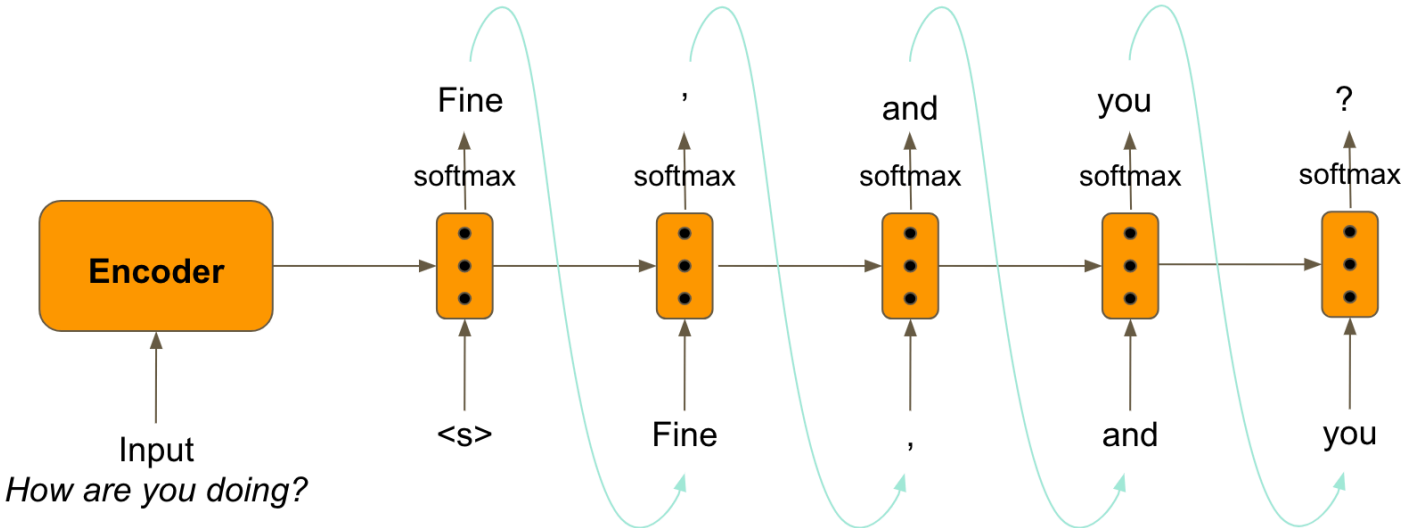


$$p(\text{and}|\text{<s> Fine,; How are you doing?})$$

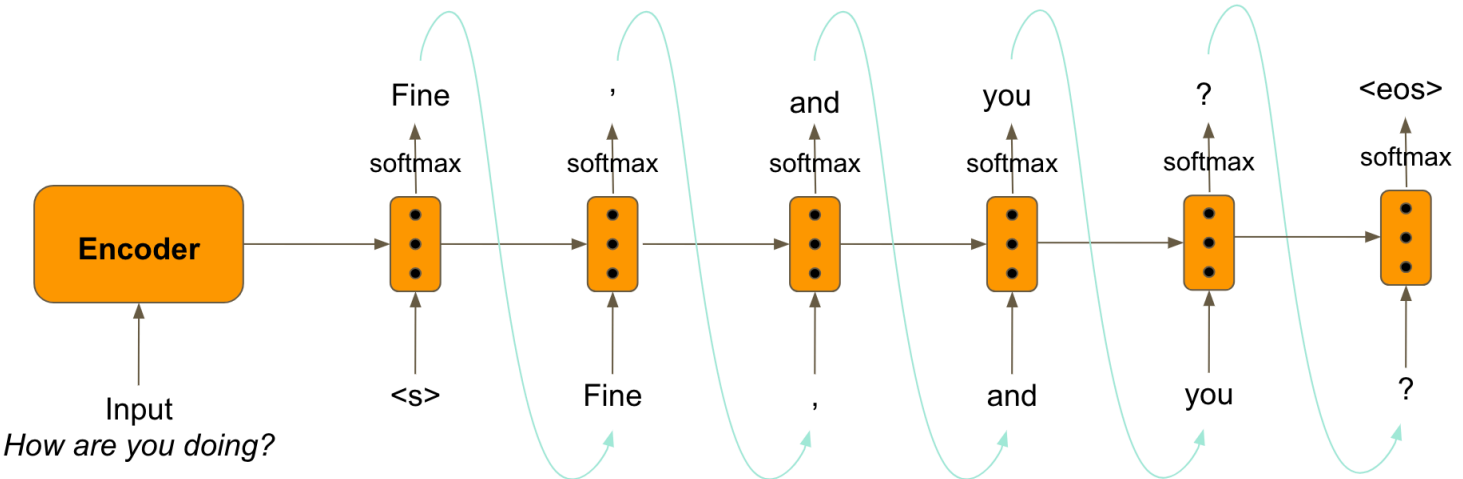
Générer du texte avec un RNN



Générer du texte avec un RNN



Générer du texte avec un RNN



2015 - Décodage avec Attention

Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

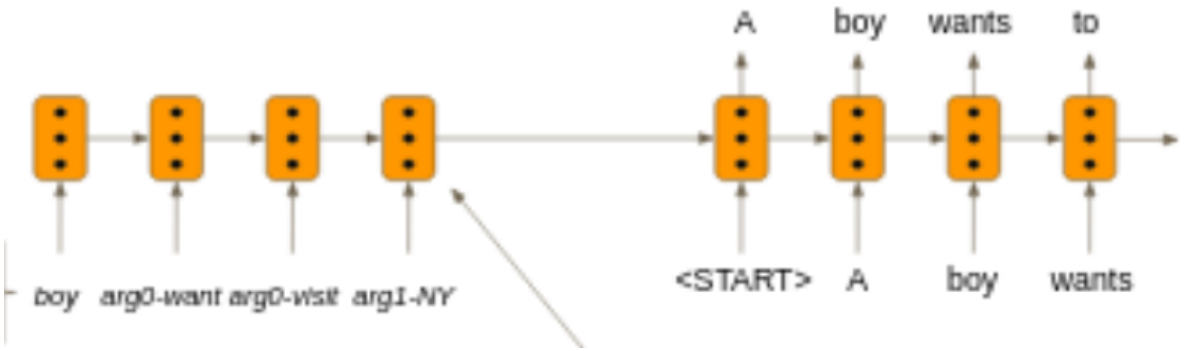
Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

ABSTRACT

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

Décodage RNN standard



- L'entrée est compressée en un vecteur de taille fixe
- Les performances diminuent avec la longueur de l'entrée

Bahdanau et al. 2015

Décodage avec attention

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

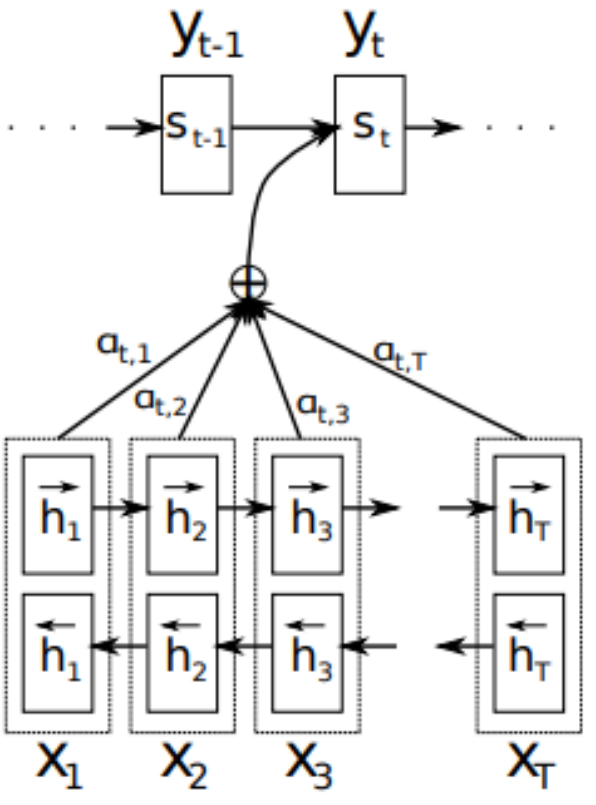
Un ***vecteur de contexte*** c_t est ajouté, qui :

- dépend des états précédents de l'encodeur et donc ***change à chaque étape***
- indique ***quelle partie de l'entrée est la plus pertinente*** pour l'étape de décodage

RNN Cross-Attention

- Un score $a_{t,j}$ est calculé entre chaque état de token d'entrée h_j et l'état précédent s_{t-1} de l'encodeur
- Le vecteur de contexte est la somme pondérée des états de l'encodeur passée à travers une couche softmax

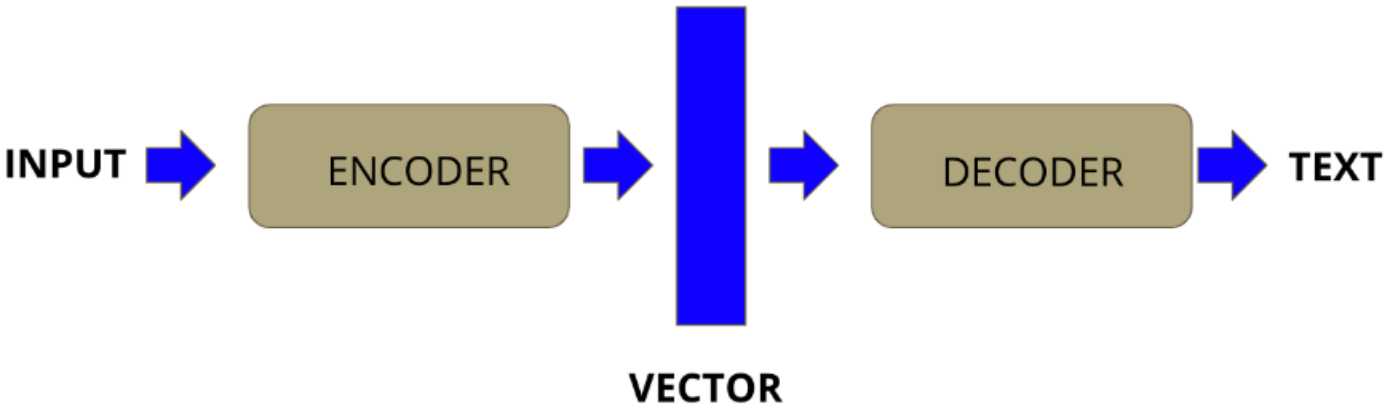
$$c_t = softmax(\sum_j \alpha_{t,j}.h_j)$$



Attention

- L'attention est une méthode permettant d'obtenir une représentation de taille fixe à partir
  - d'un ensemble arbitraire de représentations (les valeurs),
  - et d'une autre représentation (la requête)
- Encodeur-Décodeur
  - Requête = état actuel du décodeur
  - Valeurs = états cachés de l'encodeur
- Transformer
  - Requête = embedding du token
  - Valeurs = embeddings des tokens voisins

Le modèle Encodeur-Décodeur



- Encodeur : vectorise l'entrée
- Décodeur : génère de manière autorégressive à partir de cette entrée
- Attention : aide le décodeur à se concentrer sur la partie pertinente de l'entrée

2017 - Le transformeur

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

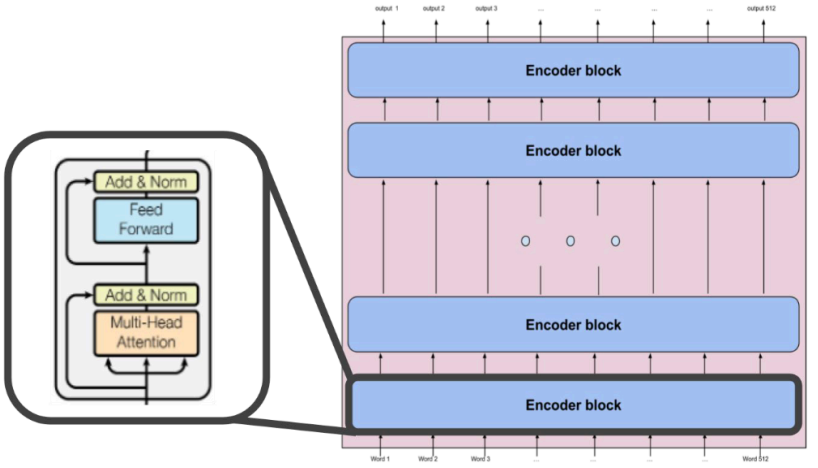
Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or

Le transformeur encodeur



- Modèle ***profond et structuré*** Empilement de blocs d'encodeur
- ***Pas de dépendances séquentielles*** (contrairement aux RNN)
- ***Auto-attention*** -- Meilleures représentations des mots
- Traitement ***parallèle*** -- Passage à échelle, pré-entrainement et ajustement fin

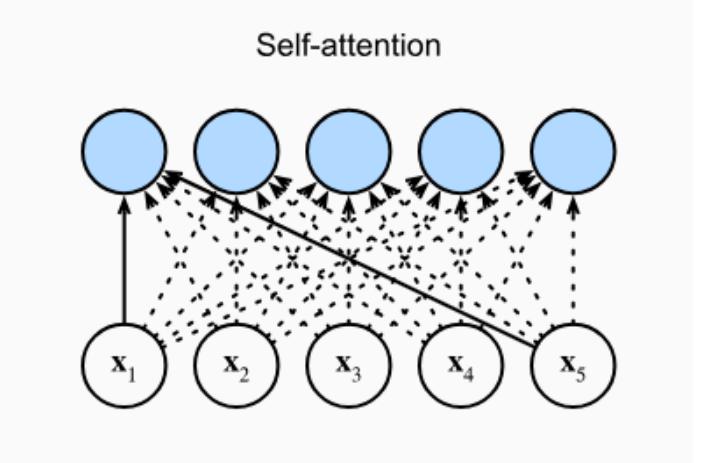
Vaswani et al. NIPS 2017

Couche d'Auto-Attention

Calcule une ***représentation dépendante du contexte de chaque mot*** dans la séquence d'entrée

- Évalue l'encodage de chaque mot d'entrée par rapport à l'encodage de chaque autre mot d'entrée
- La représentation de sortie de chaque mot est la somme pondérée des représentations de ses mots voisins

  Capture l'**ambiguïté lexicale** : le même mot aura des représentations différentes selon son contexte



Jean lit un **livre**

Jean **livre** un colis

Ce colis pèse une **livre**

La **livre** sterling est la monnaie du Royaume-Uni

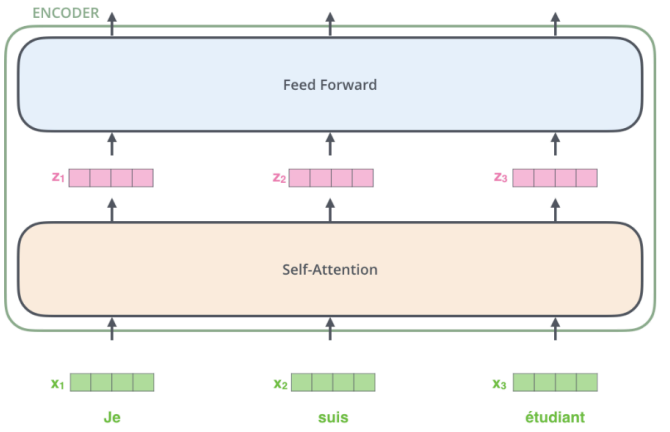
Passage à échelle

Pas de dépendances séquentielles.

Facilite le parallélisme

- Différents processeurs peuvent être utilisés pour traiter les tokens d'entrée en parallèle.

Cela peremt d'entraîner sur des quantités de données plus importantes qu'auparavant.



> *Les Transformers ont conduit à l'introduction du paradigme du **pré-entrainement avec affinage** (BERT, T5, BART) et ont facilité la création de **très grands modèles** (par exemple, ChatGPT).*

Pré-entrainement et affinage

Pré-entrainer une fois, affiner plusieurs fois

Comment ?

- Trouver une tâche (par exemple, Modélisation de Langage) pour laquelle il est facile de générer des étiquettes et pour laquelle vous pouvez obtenir de grandes quantités de données d'entraînement
- ***Pré-entrainement*** : entraîner un modèle sur ces grandes données
- ***Affinage*** : l'adapter à une tâche en utilisant des données étiquetées

Pré-entrainement et affinage - Avantages

Un modèle pré-entraîné encode beaucoup d'informations sur le langage

Données

- Moins de données étiquetées nécessaires

Efficacité

- Moins de temps pour ajuster finement que pour entraîner depuis zéro

Généralisation

- Atteint des résultats état de l'art pour une large variété de tâches : classification, inférence linguistique, similarité sémantique, réponse aux questions, etc.

2019 - BERT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

BERT

Pré-entrainement

Encodeur transformer de grande taille - 340M de paramètres, 24 couches

Pré-entraîné sur une grande quantité de texte - BooksCorpus (800M de mots) et Wikipedia en anglais (2 500M de mots)

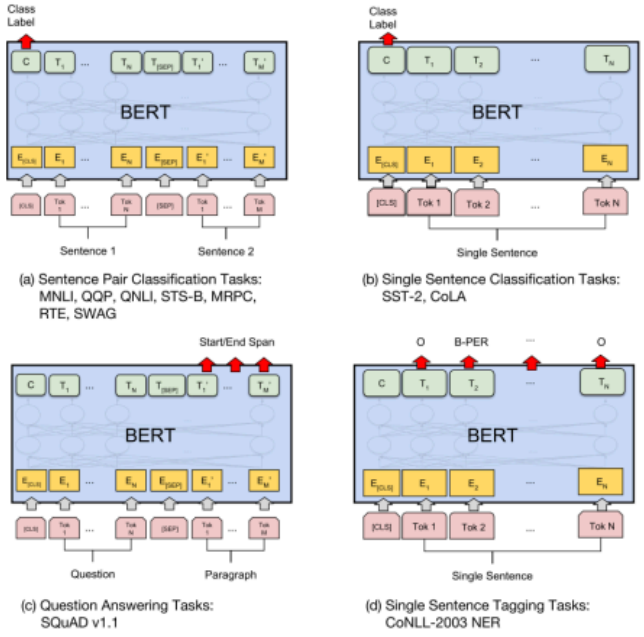
Objectif: Modéle de Langue Masqué - Prédire le mot manquant

Amélioration des représentations des mots (Self Attention)

Un modèle générique qui peut être affiné pour diverses tâches de

Affinage

Adapte les paramètres du modèle à la tâche cible en poursuivant l'entraînement sur des données étiquetées provenant de différentes tâches cibles



Impact de BERT

- Mis en open source par Google en 2018
- A atteint des ***résultats état de l'art dans 11 tâches de compréhension du langage naturel (NLU)***, y compris l'analyse des sentiments, l'étiquetage des rôles sémantiques, la classification de texte et la désambiguïsation des mots à significations multiples.
- Contrairement aux modèles précédents, tels que word2vec et GloVe, BERT traite efficacement l'***ambiguïté***, un défi majeur pour la NLU.
- ***améliore la compréhension d'environ 10 % des requêtes de recherche Google en anglais basées aux États-Unis***
  .

2018 - GPT

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com
Karthik Narasimhan OpenAI karthikn@openai.com
Tim Salimans OpenAI tim@openai.com
Ilya Sutskever OpenAI ilyasu@openai.com

Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on

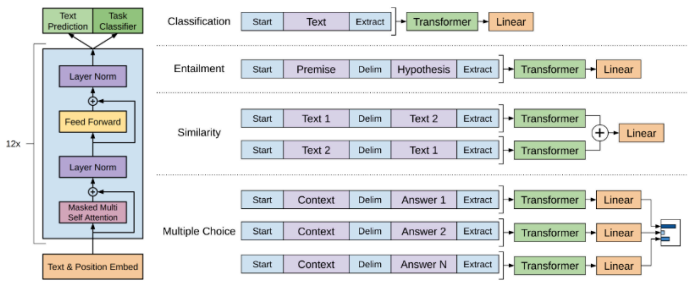
GPT - Transformer Pré-entraîné Génératif

Pré-entrainement

Grand Transformer ***décodeur*** - 117M de paramètres, 12 couches

Pré-entraîné sur un grand corpus de texte - BookCorpus 7K livres

Objectif de Modélisation de Langage - prédire le mot suivant



Affinage

GPT peut être affiné pour des tâches de compréhension (NLU)

Pendant l'affinage, le modèle possède deux têtes :

- la ***tête LM*** standard pour prédire le mot suivant
- une ***tête spécifique à la tâche*** par exemple, une tête de classification (une couche linéaire + softmax supplémentaire)

Radford et al. 2018

GPT

Améliore l'état de l'art pour 9 taches de compréhension sur 12

Results for NLI

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Amélioration de la génération de texte

GPT-2, une version plus grande (1,5B) de GPT, entraînée sur plus de données, a montré qu'elle pouvait produire du texte d'excellente qualité.

Story Generation

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

2020 - GPT3

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***

Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**

Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**

Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**

Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**

Benjamin Chess **Jack Clark** **Christopher Berner**

Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

Abstract

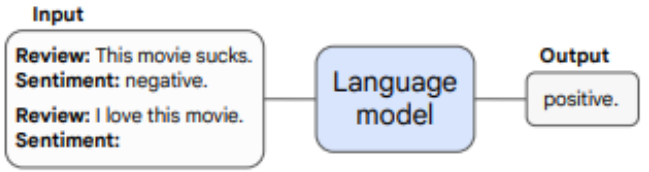
Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only

2020 - GPT3

Modèle de Langage de Grande Taille (LLM)

- Décodeur Transformer
- 175B de paramètres
- Entraîné sur 500B de mots

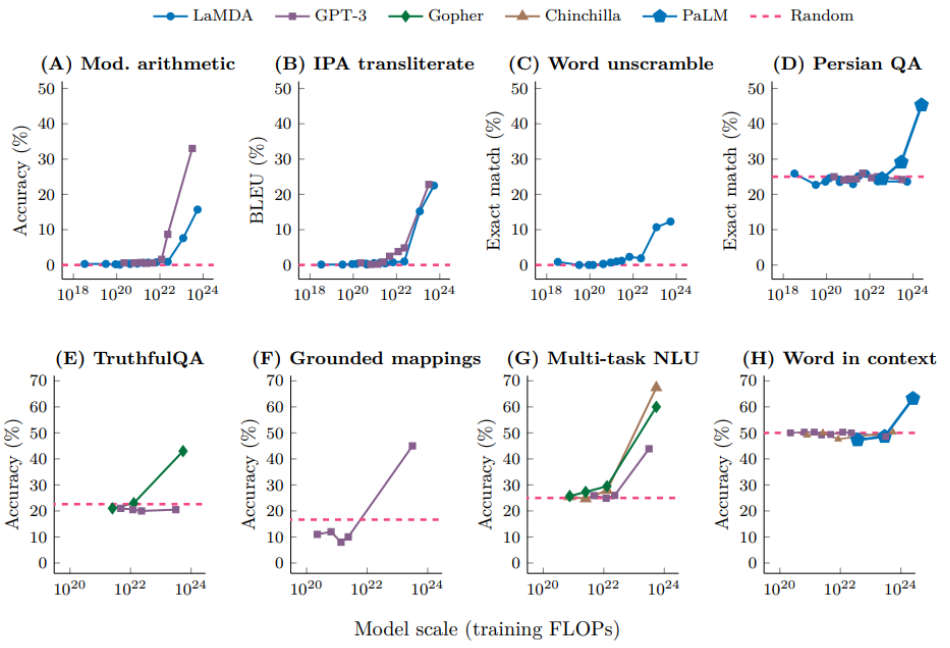
Le simple prompting suffit, pas d'affinage



La plus grande leçon que l'on peut tirer de 70 ans de recherche en IA est que les méthodes générales qui exploitent la puissance de calcul sont finalement les plus efficaces, et de loin.

Rich Sutton. "The bitter Lesson", March 2019

2020 - GPT3



Wei et al., TMLR 2022

2023 - ChatGPT et InstructGPT

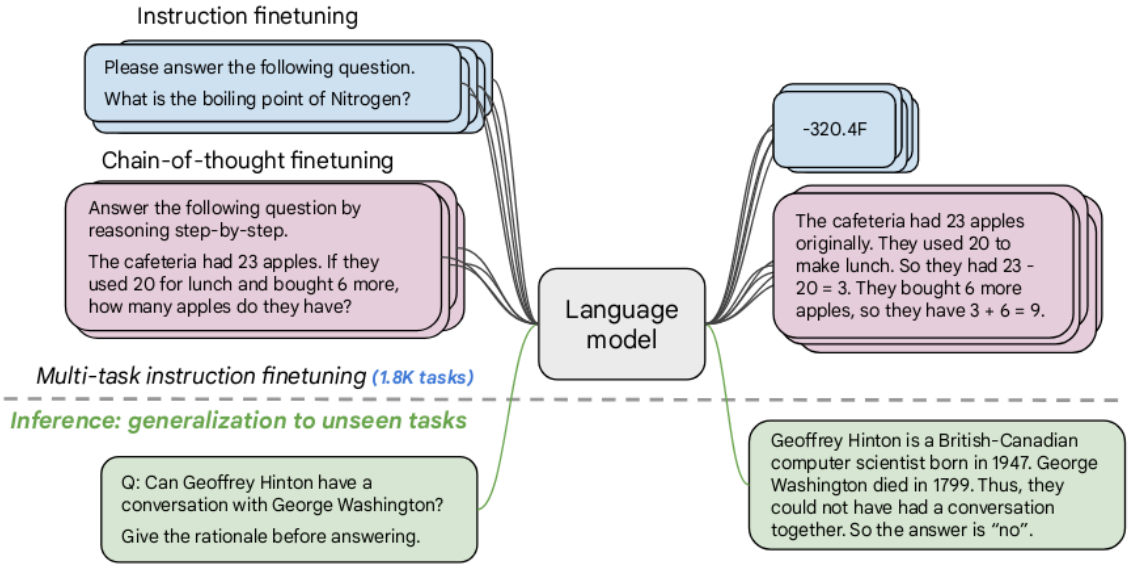
ChatGPT

- Une variante de GPT-3 optimisée pour la conversation
- Affiné sur des données de conversation
- Mieux adapté aux chatbots et à l'interaction conversationnelle

InstructGPT

- Initialisé avec GPT-3
- Affiné sur des instructions humaines
  - Ajustement supervisé sur des données spécifiques à des tâches
  - Alignement avec les préférences humaines en utilisant l'apprentissage par renforcement avec retour humain (RLHF)

Affinage par instructions



Données d'apprentissage

Le SuperNatural Instruction Dataset: 1.6K taches, 3M exemples

(a) SUP-NATINST (this work)

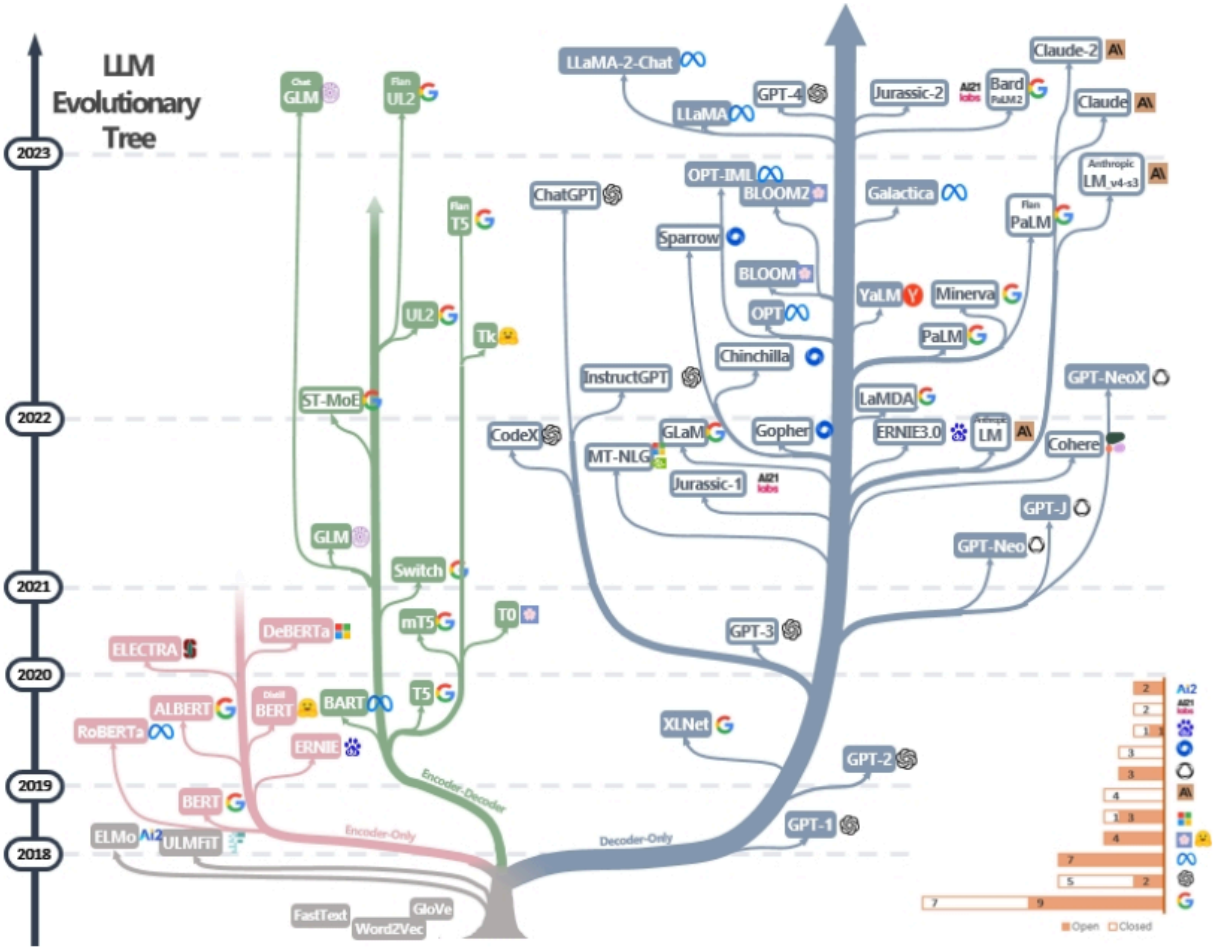
(b) NATINST

(c) PROMPTSOURCE (T0 subset)

(d) FLAN

(e) INSTRUCTGPT

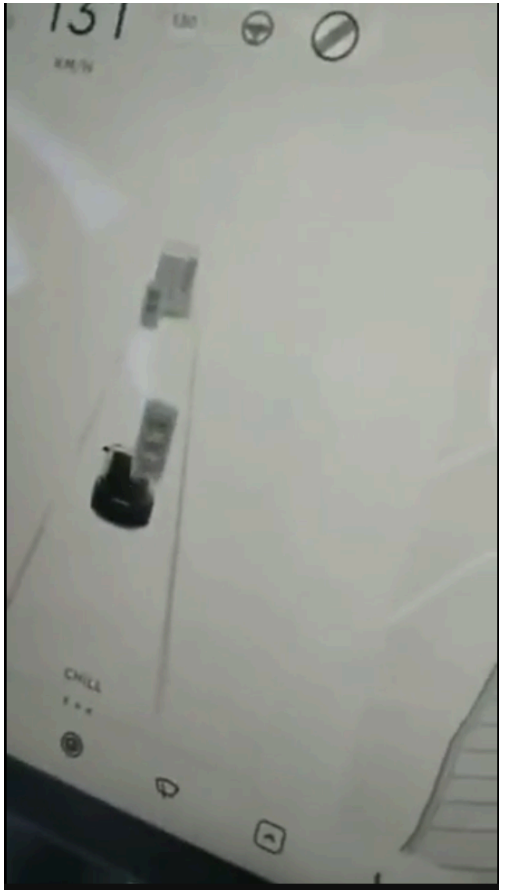
Wang et al. 2022



Yang et al. 2023, "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond". arXiv 2304.13712

Problèmes liés à la technologie

- Les LLMs font des erreurs ! Ils n'ont aucune notion de vérité.
- Biais et toxicité
- Droit d'auteur et propriété intellectuelle
- RGPD/Données personnelles
- Généralisation aux données hors domaine
  Exemple Tesla
- Coût environnemental
- Boite noire



Défis pour le TAL

Amélioration de la factualité, de la cohérence

Du générique au spécifique

- Adapter les LLMs à un nouveau domaine, une nouvelle tâche, une nouvelle langue

Évaluation

- "Facile" pour les taches de compréhension (précision, F1, etc.)
- Complexe pour les taches de génération -- problème de la ***paraphrase***

Génération à partir de données

- Vérbalisation des graphes de connaissances, données numériques, tabulaires, etc.

Multilinguisme

- Toutes les langues ne sont pas traitées de manière égale par les LLMs

Tendances

Génération Augmentée par Recherche (RAG)

- Augmente les LLMs avec des connaissances provenant de sources externes
- Hallucination, Manque d'attribution, Confidentialité des données, Contexte limité

Affinage

- Affinage efficace (LoRA, Adapters, etc.) ; Aide à l'adaptation au domaine/langue/tâche
- Apprentissage des préférences (DPO) ; Aide à la généralisation hors domaine (OOD), alignement avec les préférences humaines

IA Agentique

- Utilisation de plusieurs LLMs ensemble : Aide à la création de données, à l'évaluation

IA générative et éducation

TAL et éducation

Des applications utiles pour l'apprenant et pour l'enseignant

Résumé automatique

- Résumé de cours, texte etc.
- Synthèse d'un groupe de documents

Systèmes de Question/Réponse

- Evaluation des écrits étudiants

  > *le texte contient il les réponses à un ensemble de questions prédéfinies par l'enseignant?*

Systèmes de génération de question

- Générer des quizzes à partir d'un texte

TAL et Education

Systèmes de dialogue Humain/Machine

- Robots conversationnels et systèmes de tutorat intelligent, par ex. pour l'enseignement de la démarche d'investigation et de la démarche scientifique (Cisel et Baron, 2019)
- Systèmes adaptatifs et personnalisés, évaluation adaptative et correction automatique

Recherche d'Information

- Recommandation et sélection de contenus, de ressources personalisés

Calcul de similarité sémantique entre deux textes

- Détection de plagiat

...

Atelier

Comment interagir avec les LLMs ?

A distance

- Sur une interface Web
- Sur le Cloud à travers une API (Application Programming Interface): Groq, Together.ai, Colab

En local

- Ollama
- La librairie Transformers de Huggingface

A distance sur une interface web

Groq

Texte

Henri Poincaré est le fils d'Émile Léon Poincaré, doyen de la faculté de médecine de Nancy, et de son épouse Marie Pierrette Eugénie Launois. Il est le neveu d'Antoni Poincaré, ce qui en fait le cousin germain des fils de ce dernier : l'homme politique et président de la République française Raymond Poincaré et Lucien Poincaré, directeur de l'Enseignement secondaire au ministère de l'Instruction publique et des Beaux-Arts. La sœur d'Henri, Aline Poincaré, a épousé le philosophe Émile Boutroux.

À cinq ans, il contracte la diphtérie, le laissant paralysé durant cinq mois, ce qui l'incite à se plonger dans la lecture.

Élève d'exception au lycée impérial de Nancy, il obtient le 5 août 1871, le baccalauréat en lettres, mention « Bien », et le 7 novembre 1871 son baccalauréat en sciences, où il faillit être refusé à cause d'un zéro en composition de mathématiques. Il semblerait qu'il soit arrivé en retard et ait mal compris le sujet, un problème sur les séries convergentes, domaine dans lequel il apportera des contributions importantes. Mais il se rattrape brillamment à l'oral et est finalement admis avec une mention « Assez Bien »

System prompt "You are an assistant who generates questions in French from a text in French."

User prompt "Generate a single question from the following text, delimited by ###:"

A distance à travers une API

Code

En local avec Ollama

Ollama est une librairie pour le déploiement en local.

Elle permet de démarrer un serveur compatible avec le protocole OpenAI.

Pour commencer il suffit d'installer Ollama via ce lien.

On peut accéder à divers models: Model Library

```
# Pour utiliser un modèle:
ollama pull gemma:2b
```

```
>>> Enabling and starting ollama service...
>>> NVIDIA GPU installed.
(base) claire@karani:~/inriagit/slides/remarks/nb-genai-educ$ ollama run gemma:2B
pulling manifest
pulling c1864a5eb193... 100%                                    1.7 GB
pulling 097a36493f71... 100%                                    8.4 KB
pulling 109037bec39c... 100%                                    136 B
pulling 22a838ceb7fb... 100%                                    84 B
pulling 887433b89a90... 100%                                    483 B
verifying sha256 digest
writing manifest
success
>>> Qui est Henri Poincaré ?
Henri Poincaré est un scientifique et philosophe français connu pour ses recherches sur les géodesicaux,
les géophysique et la logique. Il est considéré comme l'un des plus influents scientifiques du XXe
siècle.
```

En local avec Ollama et Python

On installe la librairie Python Ollama

```
! pip install ollama
import ollama

response = ollama.generate(model='gemma:2b',prompt='what is a qubit?')
print(response['response'])
```

Code: local-ollama.ipynb)

En local avec Huggingface et Python

Code

Code

Questions ?

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICLR.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei. Language Models are Few-Shot Learners. arXiv:2005.14165, Jul 2020.

Deng, Guoqiang, Min Tang, Yuhao Zhang, Ying Huang, and Xuefeng Duan. 2022. "Privacy-Preserving Outsourced Artificial Neural Network Training for Secure Image Classification" Applied Sciences 12, no. 24: 12873. https://doi.org/10.3390/app122412873

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long- short-term memory. Neural computation 9(8):1735–1780

Kyunghyun Cho, Bart Van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the Properties of Neural Machine Translation: Encoder-decoder Approaches. In Proc. of SSST.

Radford et al. 2018. Improving Language Understanding by Generative Pre-Training

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533--536.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks.. In Proc. of NIPS. pages 3104–3112

Rich Sutton, March 2019. The bitter Lesson]

Vaswani et al. NIPS 2017. Attention is all you need]

Wang et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks.]

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean and William Fedus. Emergent Abilities of Large Language Models. Transactions on Machine Learning Research (08/2022)

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language Models Are Zero-Shot Learners. ICLR 2022.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le and Jason Wei. Scaling Instruction-Finetuned Language Models. arXiv, 2022.