

DU PROMPTING AU FINE-TUNING

Vincent Guigue & Rémy Découpes
vincent.guigue@agroparistech.fr

TEXTUAL DATA

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

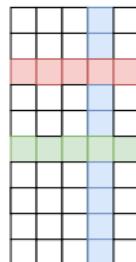
DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

DATA THAT CAN BE READ AND MANIPULATED AS TEXT

From tabular data to text

- Tabular data
 - Fixed dimension
 - Continuous values

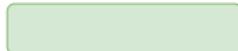
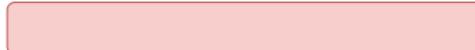


$$\rightarrow f(\boxed{\quad\quad\quad}) = \text{pred}$$

- Textual data
 - Variable length
 - Discrete values

this new iPhone, what a marvel

An iPhone? What a scam!

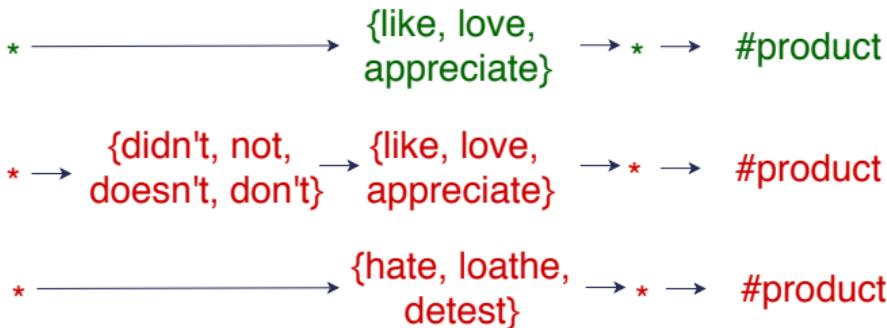


AI + Textual Data: Natural Language Processing (NLP)

NLP = largest scientific community in AI

Linguistics [1960-2010]

Rule-based Systems:



- Requires expert knowledge
- Rule extraction ⇔ very clean data
- Very high precision
- Low recall
- Interpretable system



AI + Textual Data: Natural Language Processing (NLP)

NLP = largest scientific community in AI

Machine Learning [1990-2015]



AI + Textual Data: Natural Language Processing (NLP)

NLP = largest scientific community in AI

Linguistics [1960-2010]

- Requires expert knowledge
- Rule extraction \Leftrightarrow
very clean data
- + Interpretable system
- + Very high precision
- Low recall

Machine Learning [1990-2015]

- Little expert knowledge needed
- Statistical extraction \Leftrightarrow
robust to noisy data
- ≈ Less interpretable system
- Lower precision
- + Better recall

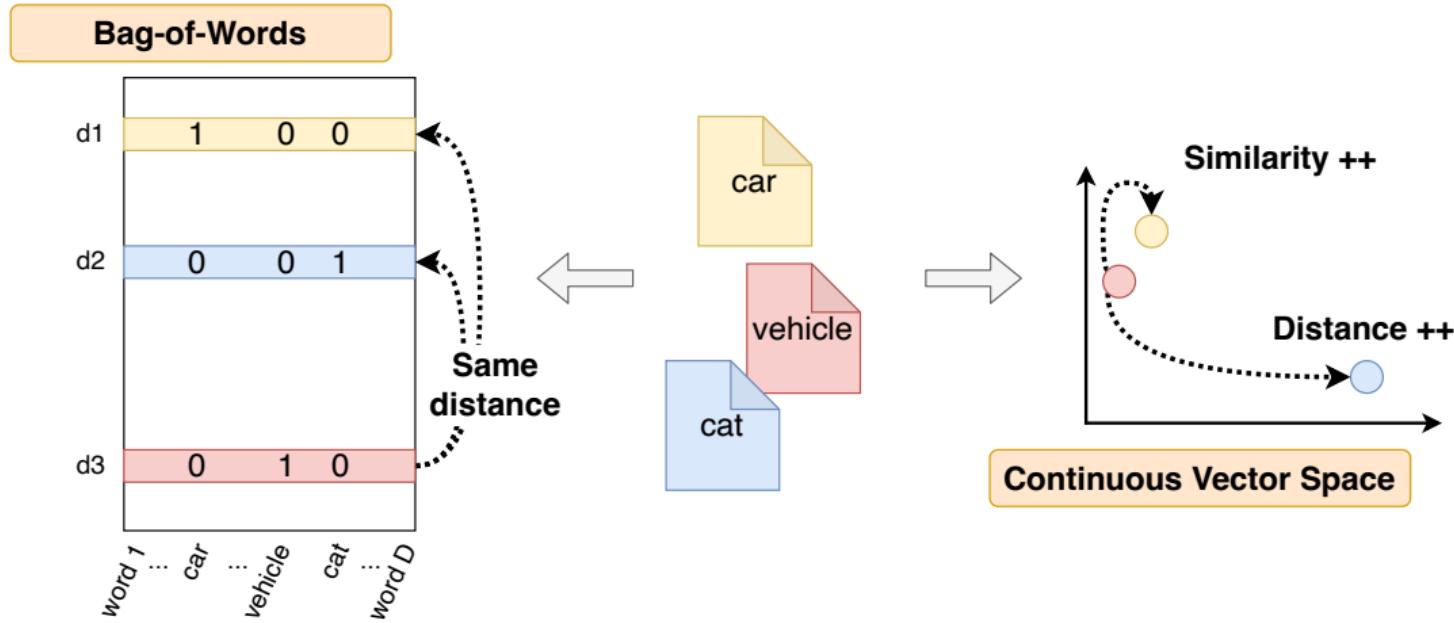
Precision = criterion for acceptance by industry

→ Link to metrics

Deep/Representation Learning for Text Data

From Bag of Words to Vector Representations

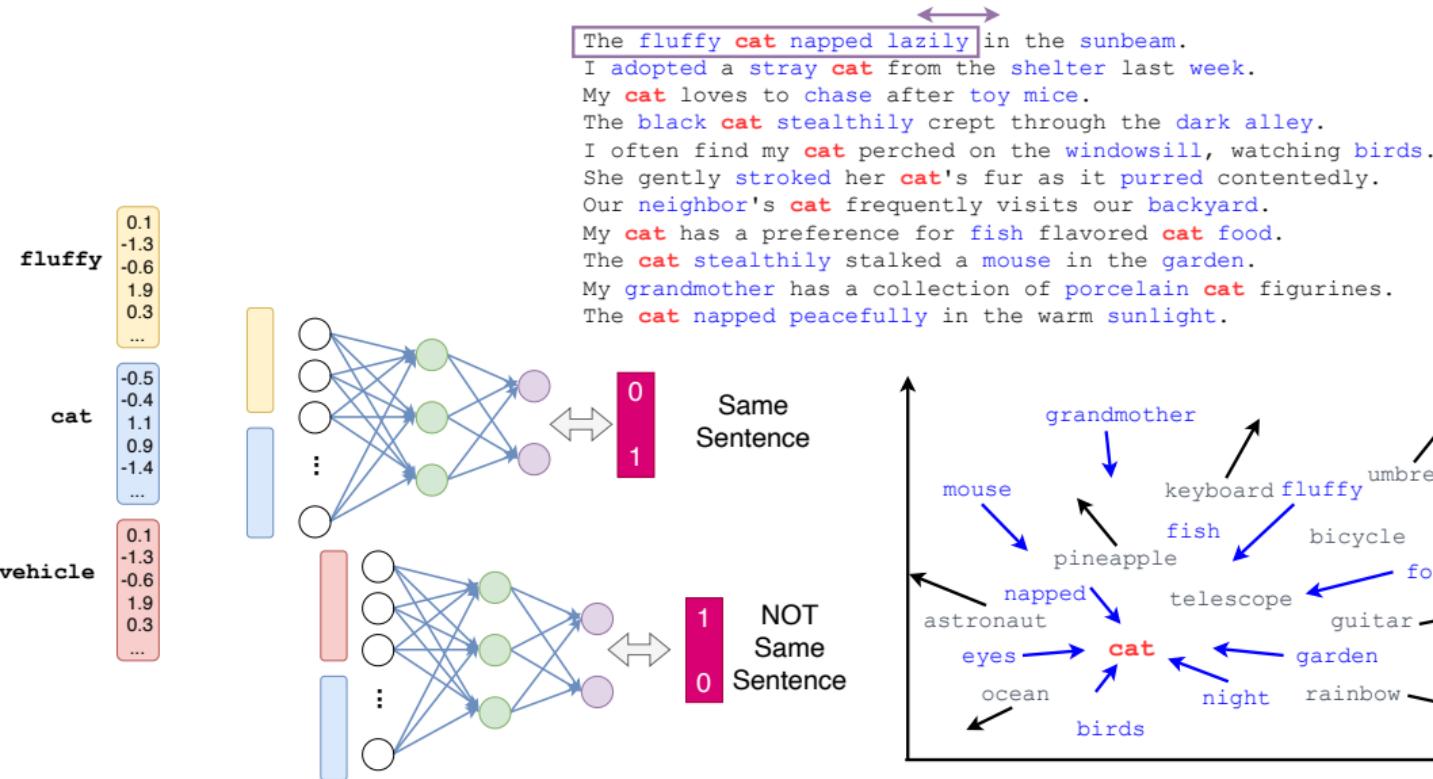
[2008, 2013, 2016]



Deep/Representation Learning for Text Data

From Bag of Words to Vector Representations

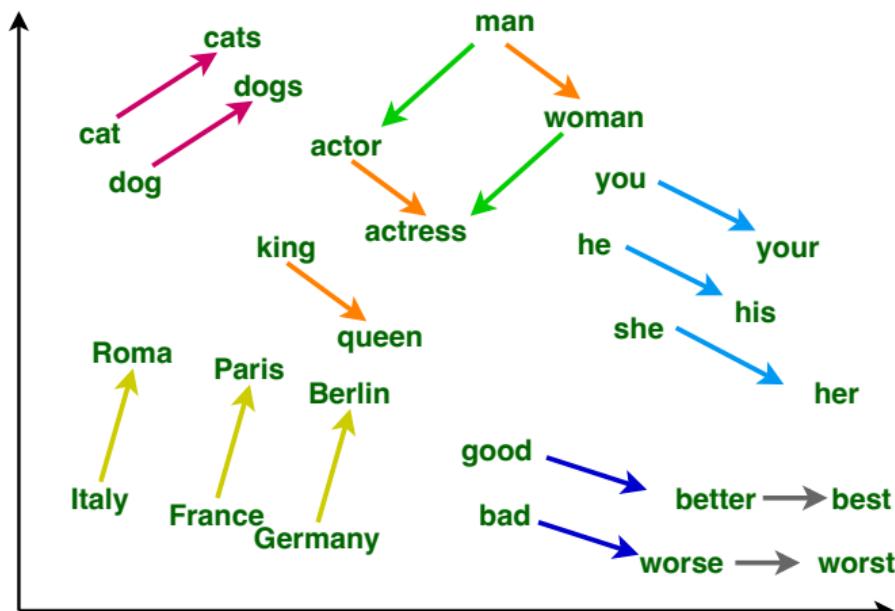
[2008, 2013, 2016]



Deep/Representation Learning for Text Data

From Bag of Words to Vector Representations

[2008, 2013, 2016]



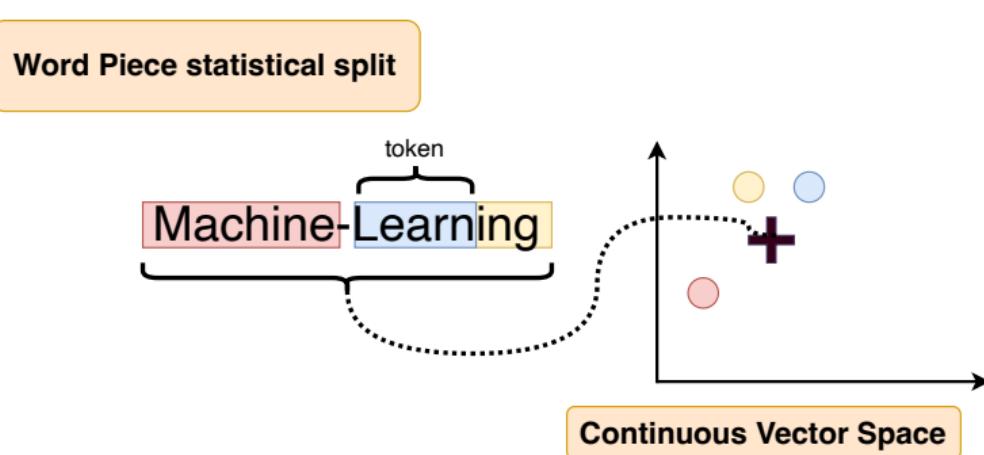
- Semantic Space:
similar meanings
↔
close positions
- Structured Space:
grammatical regularities,
basic knowledge, ...

Deep/Representation Learning for Text Data

From Bag of Words to Vector Representations

[2008, 2013, 2016]

From Words to Tokens

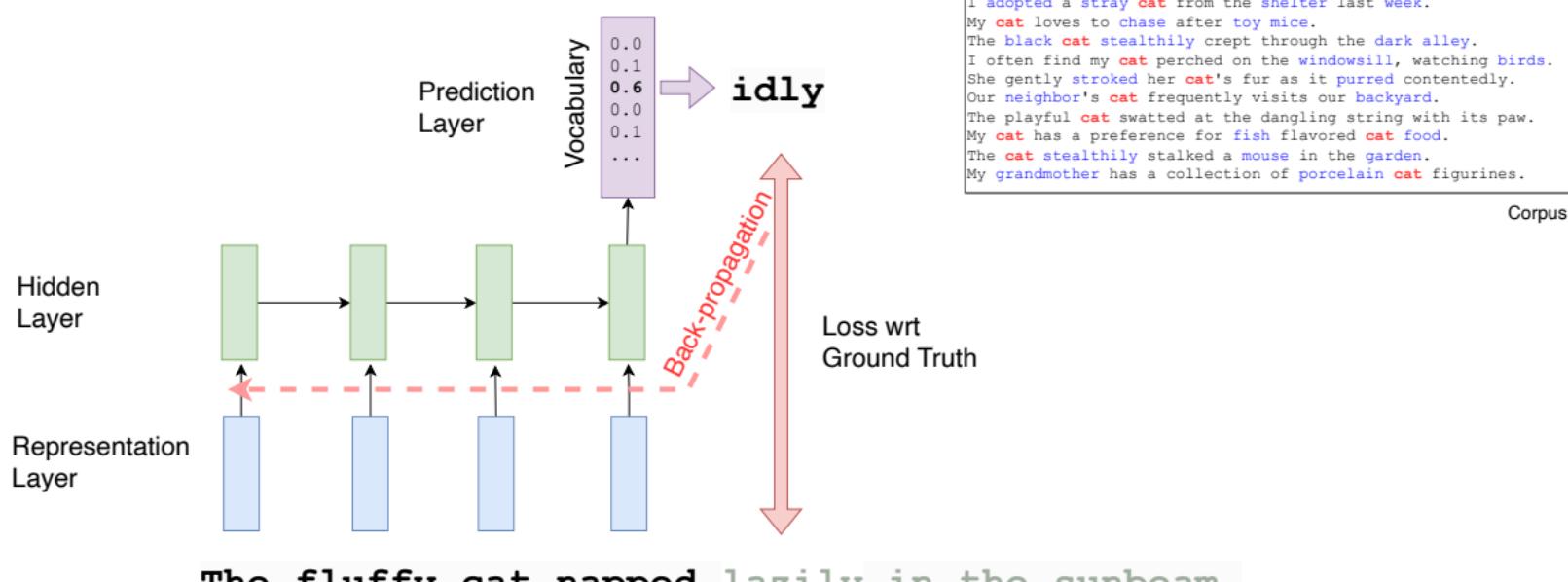


- Representation of unknown words
- Adaptation to technical domains
- Resistance to spelling errors

Enriching word vectors with subword information. [Bojanowski et al. TACL 2017.](#)

Aggregating word representations: towards generative AI

- Generation & Representation
- New way of learning word positions

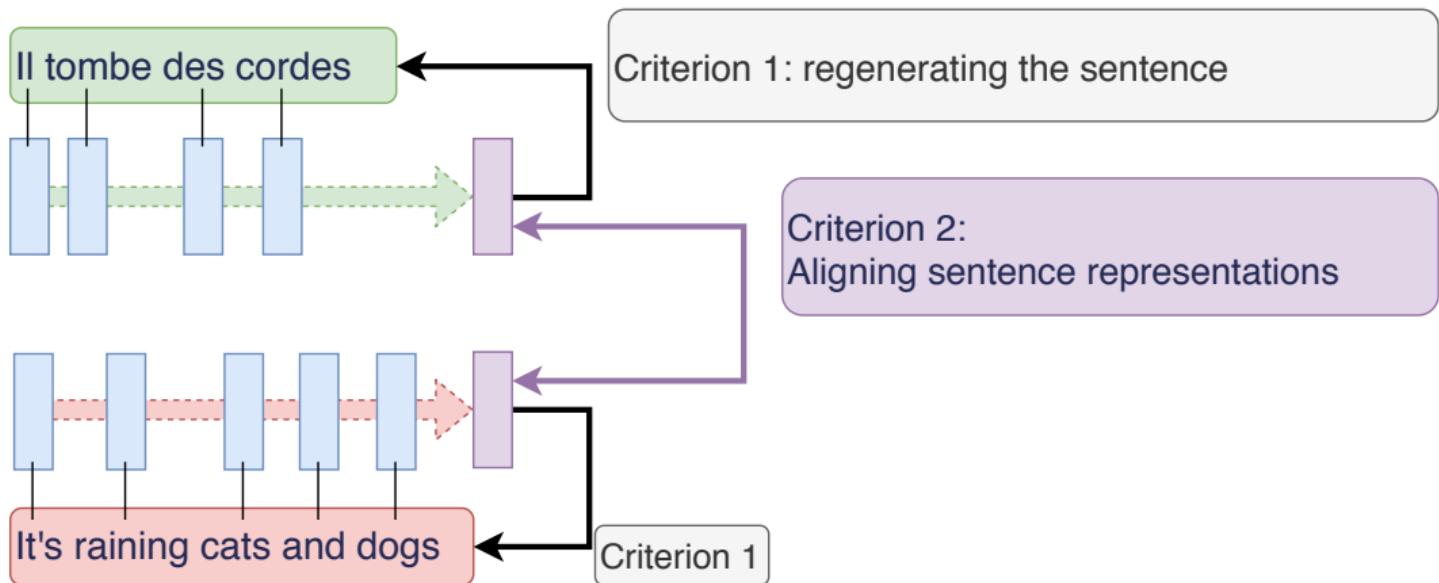


Use-Case: Machine Translation



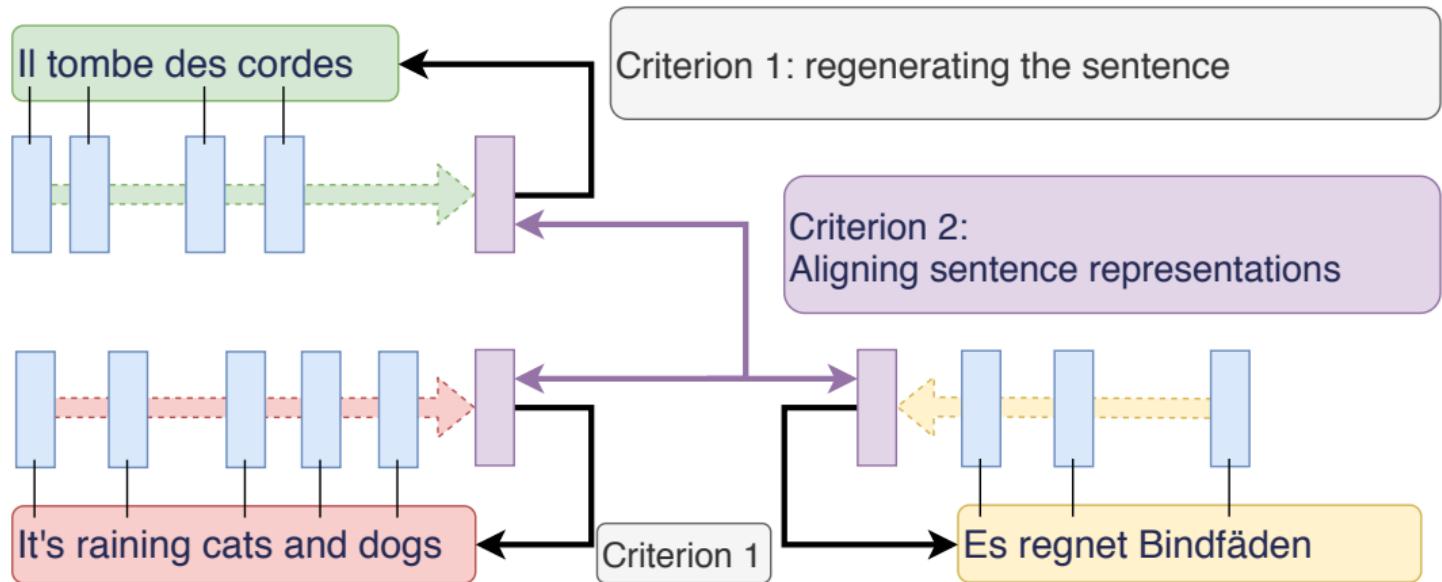
Beyond word-for-word translation, multilingual representation of sentences

Use-Case: Machine Translation



Beyond word-for-word translation, multilingual representation of sentences

Use-Case: Machine Translation



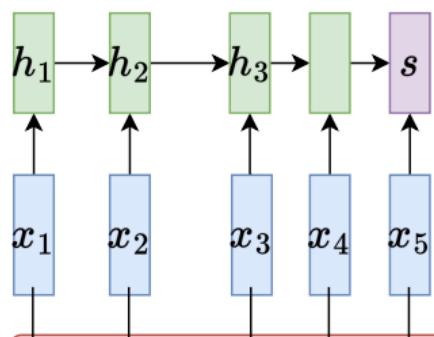
Beyond word-for-word translation, multilingual representation of sentences

A

Transformer architecture: state-of-the-art aggregation

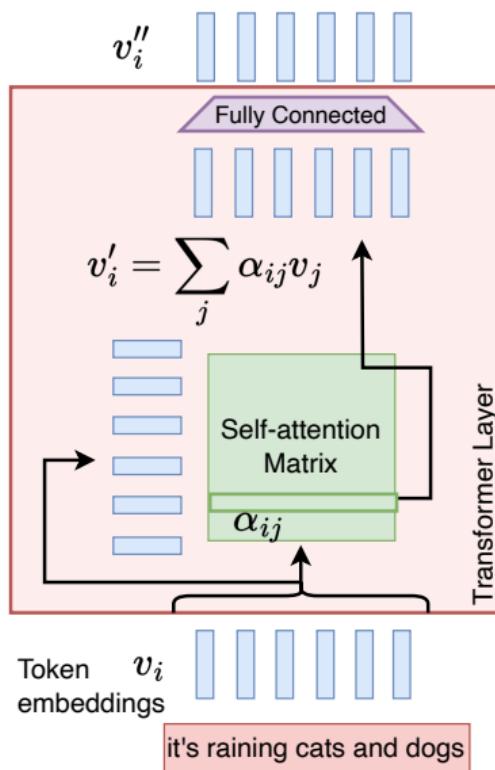
Recurrent Neural Network:

$$h_{t+1} = h_t W_1 + x_{t+1} W_2$$



It's raining cats and dogs

Transformer:



Attention is all you need, [Vaswani et al. NeurIPS 2017](#)

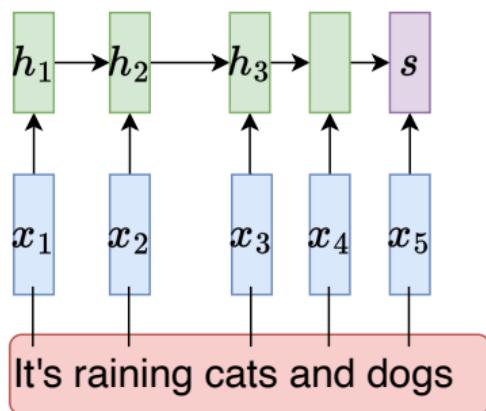
Sequence to Sequence Learning with Neural Networks, [Sutskever et al. NeurIPS 2014](#)

A

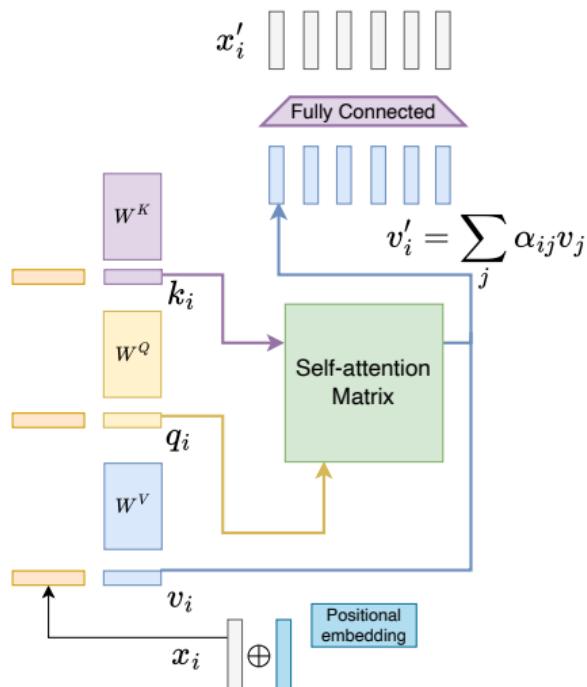
Transformer architecture: state-of-the-art aggregation

Recurrent Neural Network:

$$h_{t+1} = h_t W_1 + x_{t+1} W_2$$



Transformer:



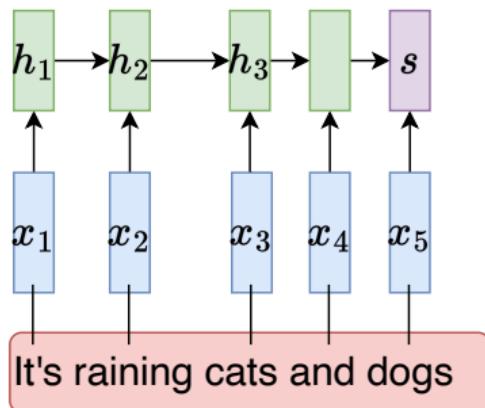
Attention is all you need, [Vaswani et al. NeurIPS 2017](#)

Sequence to Sequence Learning with Neural Networks, [Sutskever et al. NeurIPS 2014](#)

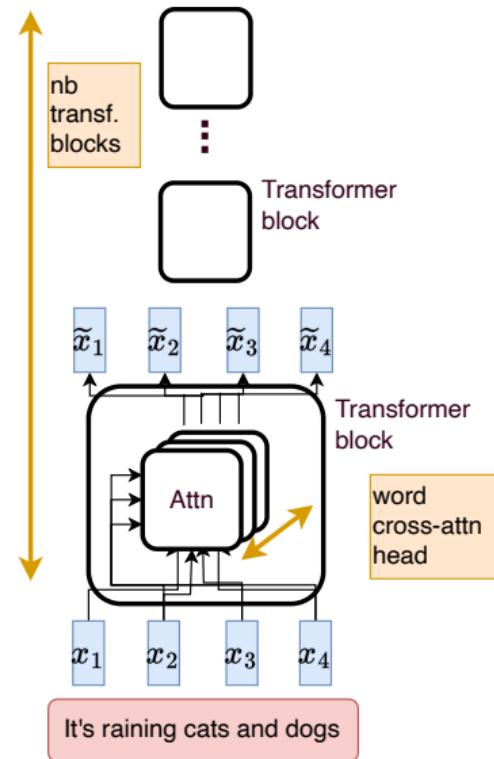
Transformer architecture: state-of-the-art aggregation

Recurrent Neural Network:

$$h_{t+1} = h_t W_1 + x_{t+1} W_2$$



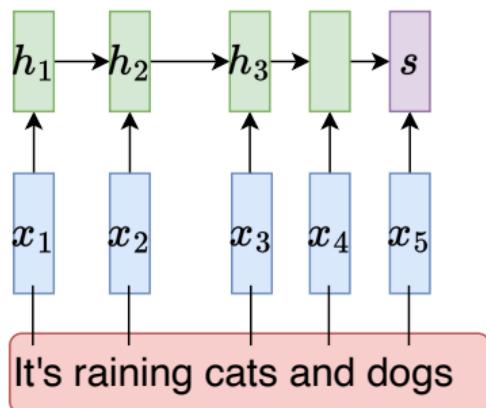
Transformer:



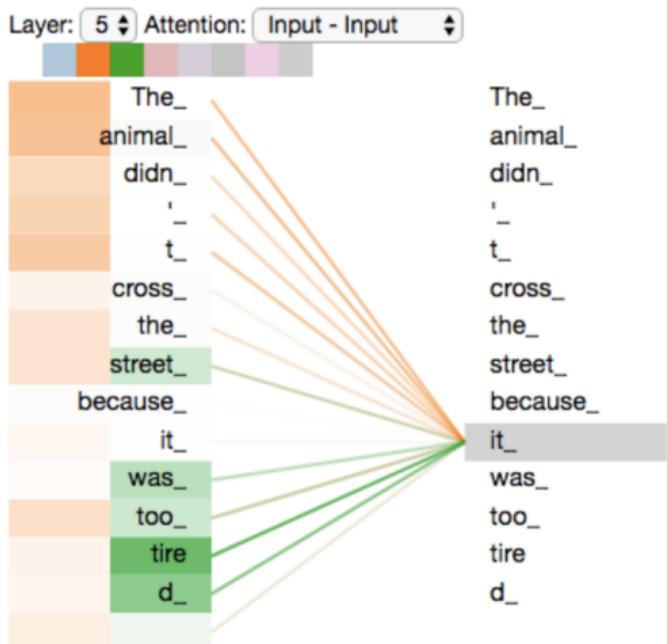
Transformer architecture: state-of-the-art aggregation

Recurrent Neural Network:

$$h_{t+1} = h_t W_1 + x_{t+1} W_2$$



Transformer:

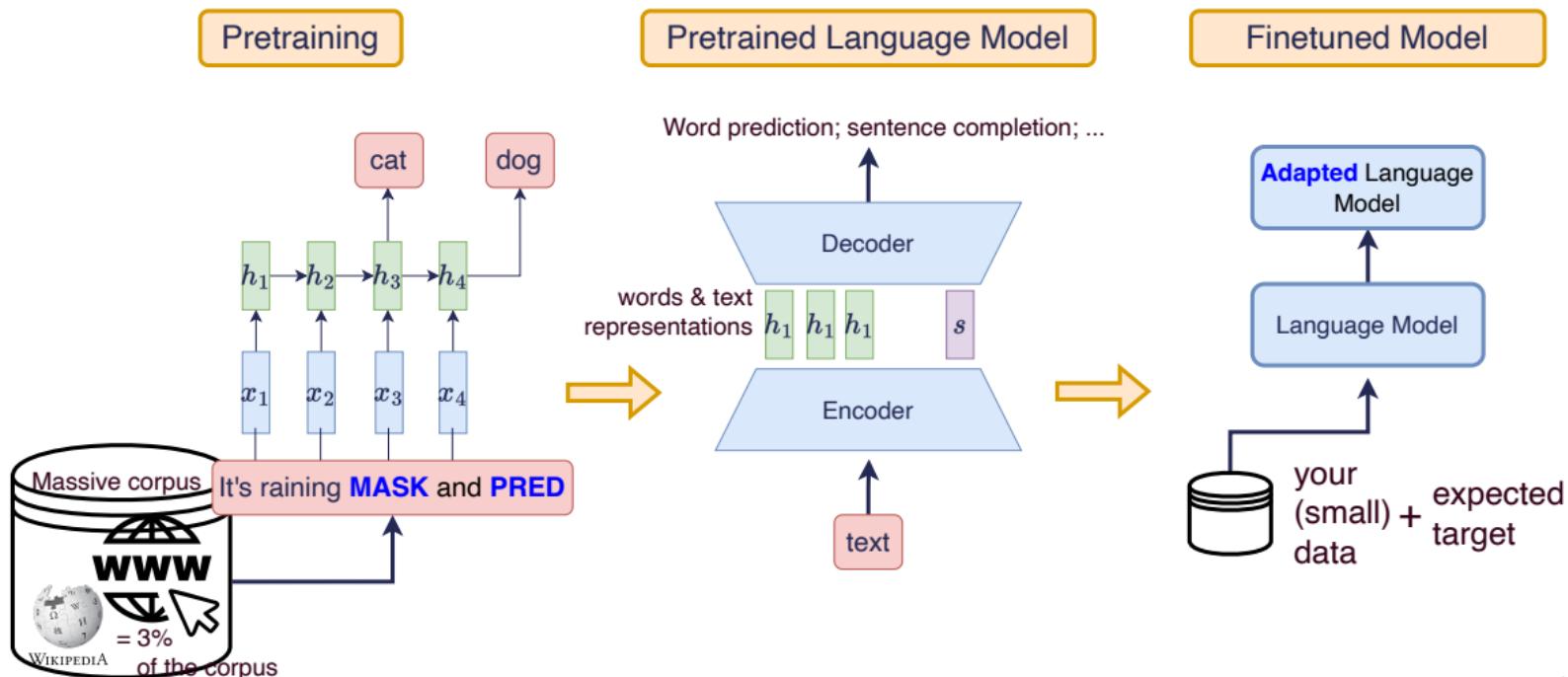


Attention is all you need, [Vaswani et al. NeurIPS 2017](#)

Sequence to Sequence Learning with Neural Networks, [Sutskever et al. NeurIPS 2014](#)

A new developpement paradigm since 2015

- Huge dataset + huge archi. \Rightarrow unreasonable training cost
- Pre-trained architecture + 0-shot / finetuning

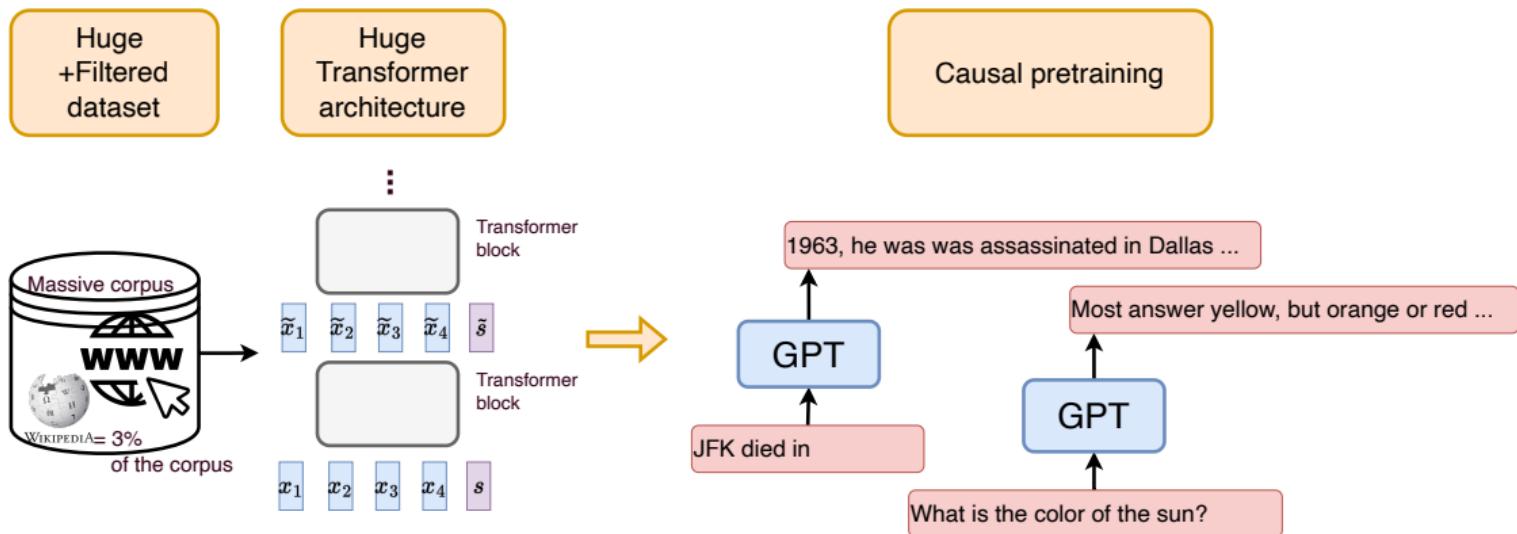


GENERATIVE AI & FINETUNING



The genAI wave

- *Causal* architecture / generating relevant answers



- Grammatical skills: singular/plural agreement, tense concordance
- Knowledges: entities, names, dates, places



A revolution in NLP: new formulation, new(?) metrics

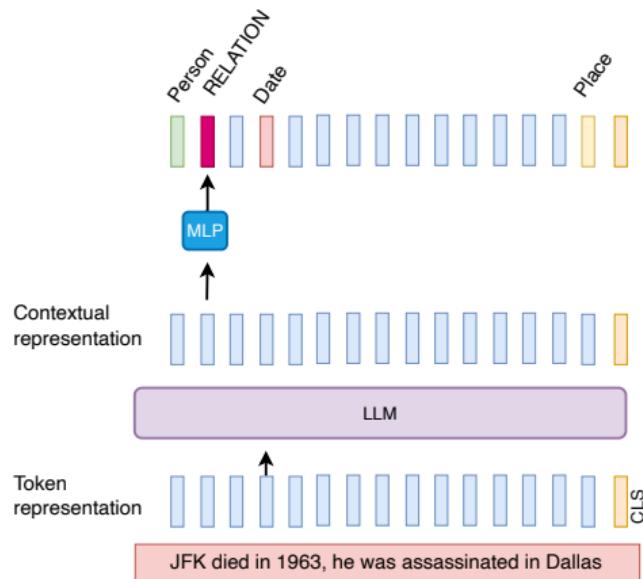
- input = text... output = text also !!!

Find entities in
JFK was assassinated in Dallas in 1963



Here are the entities
JFK → Person
Dallas → Location (City)
1963 → Date/Year

LLM



- How to evaluate Entity location?
- What about rephrasing?
- Over detection

JFK PERSON was assassinated in Dallas GPE in 1963 DATE

Formatting constraints

Find entities in

JFK was assassinated in
Dallas in 1963



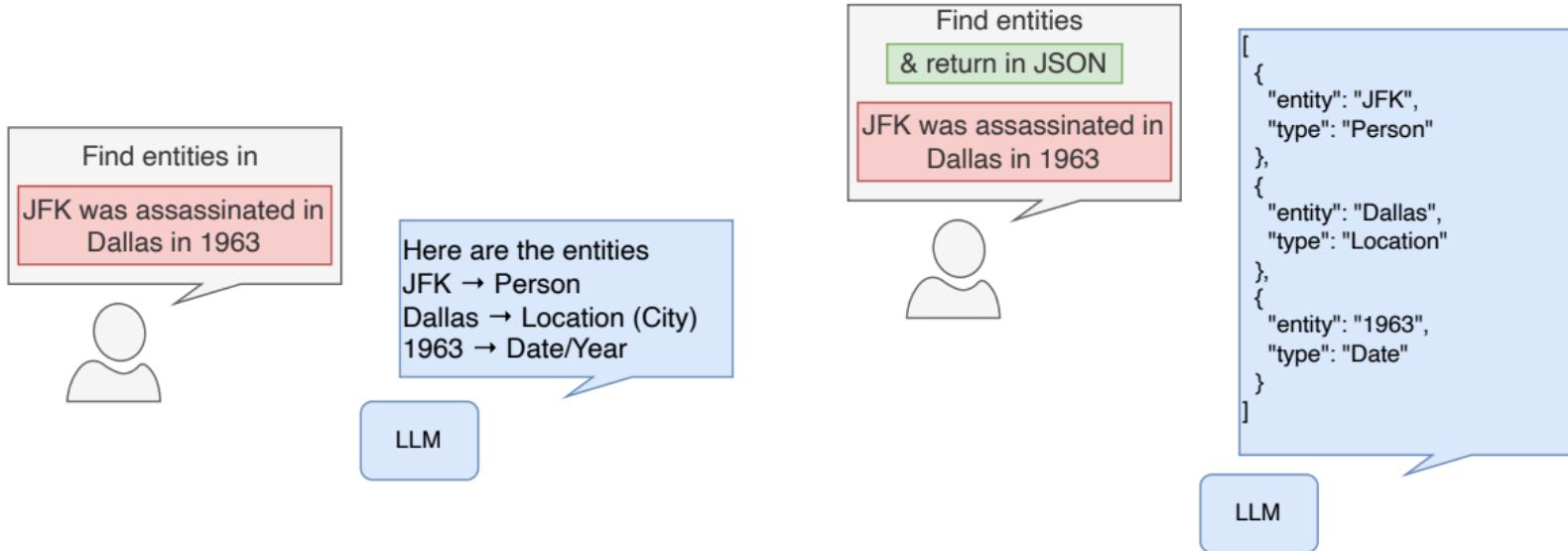
Here are the entities
JFK → Person
Dallas → Location (City)
1963 → Date/Year

LLM

- Prompt optimization: role, task, categories, examples, ...



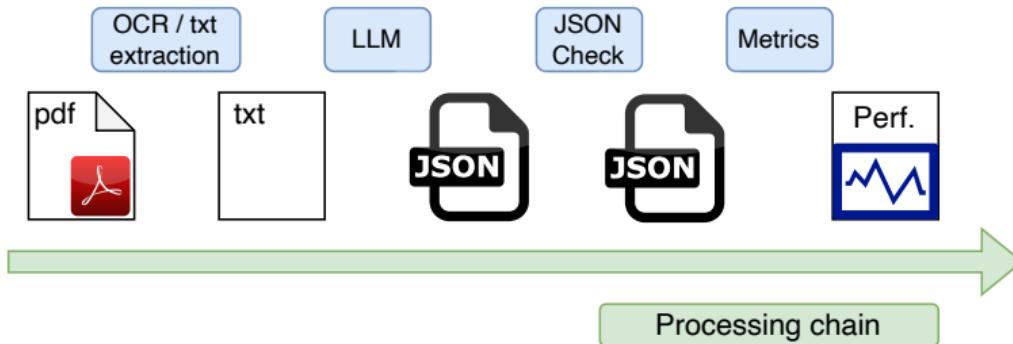
Formatting constraints



- Prompt optimization: role, task, categories, examples, ...
- Adding constraints ⇒ processing chain
- Forcing the system to return JSON... **and only JSON**



Processing chain



- Famous framework = langchain
- Challenge: all components should be replacable
- OCR : pypdf, tesseract...
- LLM : ollama, HuggingFace, GPT, ...
- JSON : regex, LLM constraint, multiple queries
- Metrics

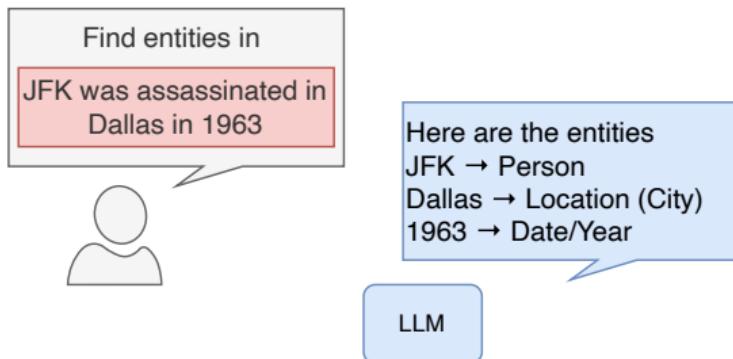


Zero-shot

Zero-shot

Exploiting a model on a task...

(almost) without training



- Is the task really new?
 - to be efficient, LLM are trained on multiple instructions
 - Risk of data contamination
- Which performance to expect?



1st optimization step: Prompting

Prompting = the way to ask a question

⇒ impact on the performance

You are an information extraction system. Your task is to read an English news article and extract named entities of the following types:

- **PERSON**: names of individual people
- **LOCATION**: geographical entities such as cities, countries, landmarks
- **ORGANIZATION**: companies, institutions, government bodies, media outlets, NGOs, etc.
- **DATE**: explicit calendar dates or temporal expressions (e.g., "January 5, 2023", "last Monday")

Return the results in **JSON format** with the following structure:

json

Copier le code

```
{  
    "PERSON": [...],  
    "LOCATION": [...],  
    "ORGANIZATION": [...],  
    "DATE": [...]  
}
```

If a category has no entities, return an empty list. Do not include duplicates.

Text to analyze:



1st optimization step: Prompting

Prompting = the way to ask a question

⇒ impact on the performance

You are an information extraction system. Your task is to read an English news article and extract named entities of the following types:

- **PERSON**: names of individual people
- **LOCATION**: geographical entities such as cities, countries, landmarks
- **ORGANIZATION**: companies, institutions, government bodies, media outlets, NGOs, etc.
- **DATE**: explicit calendar dates or temporal expressions (e.g., "January 5, 2023", "last Monday")

Return the results in **JSON format** with the following structure:

json

Copier le code

```
{  
    "PERSON": [...],  
    "LOCATION": [...],  
    "ORGANIZATION": [...],  
    "DATE": [...]  
}
```

ROLE

TASK

Class DEF

Format

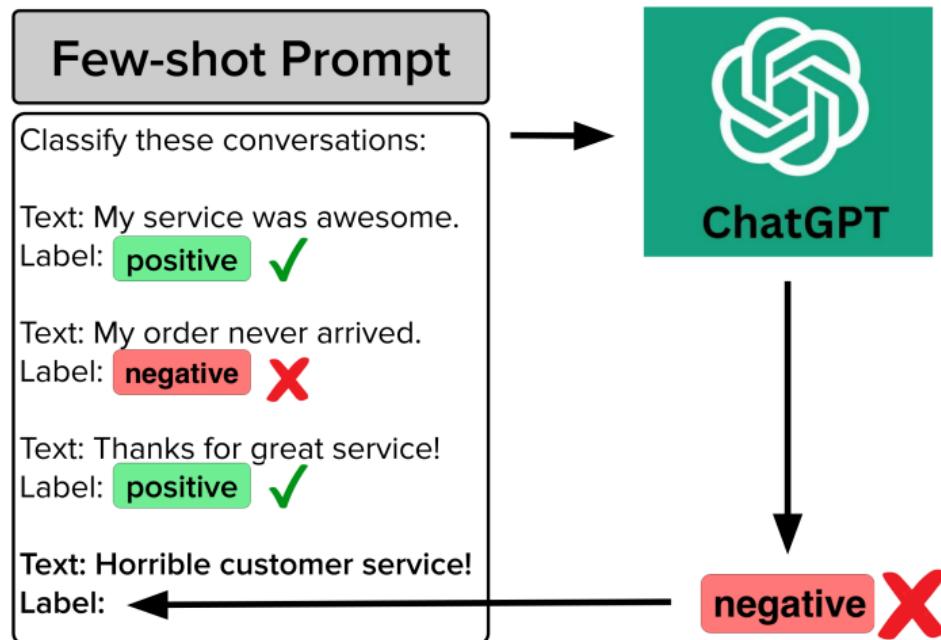
Supp.

If a category has no entities, return an empty list. Do not include duplicates.



Few-shot learning (no optimization yet!)

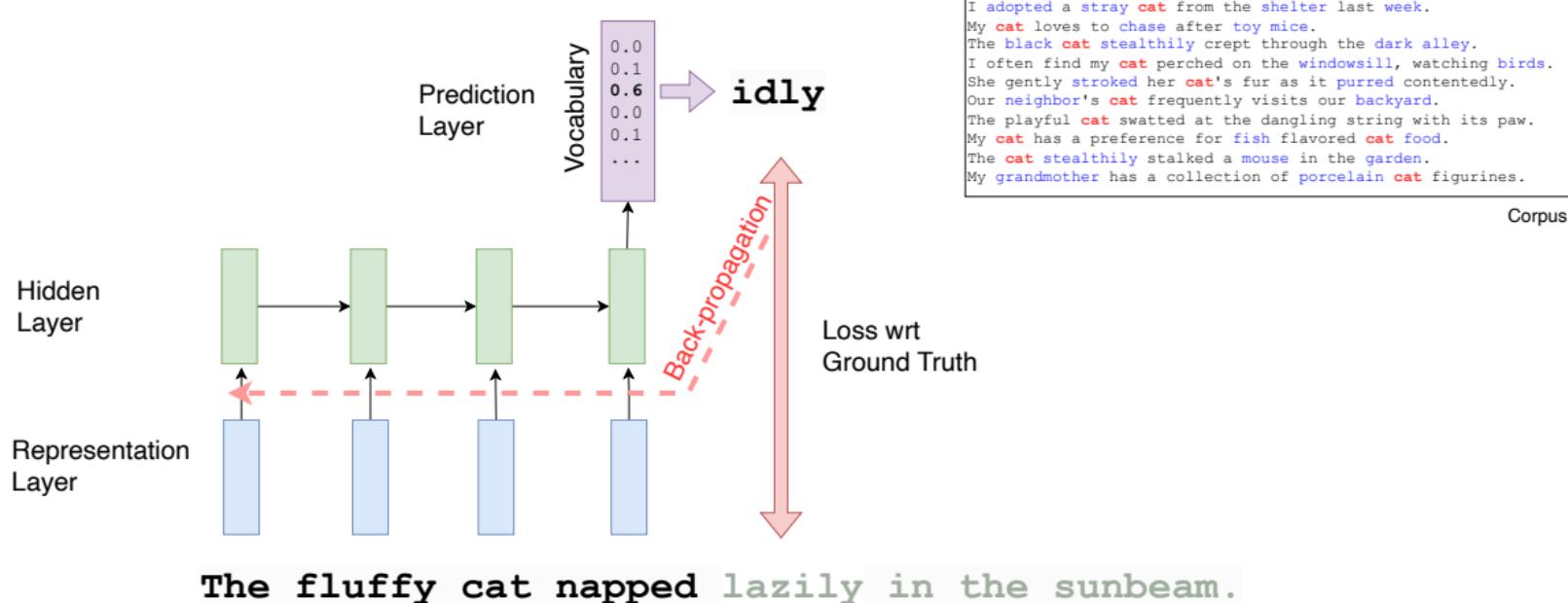
Give (input ⇔ output) on few examples to guide the extraction





Finetuning of genAI

Most generative finetuning = train to generate the ground truth

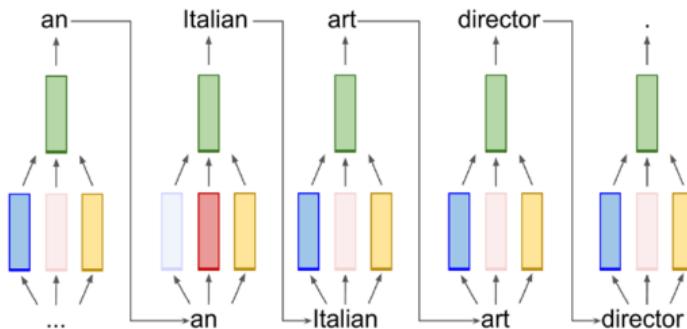




Faithfulness optimization : At the token level¹

| | |
|--------------|------------------|
| Name | Giuseppe Mariani |
| Occupation | Art director |
| Years active | 1952 - 1992 |

Giuseppe Mariani was an **Italian** art director.



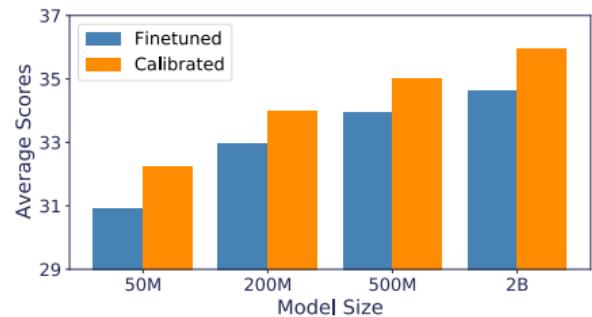
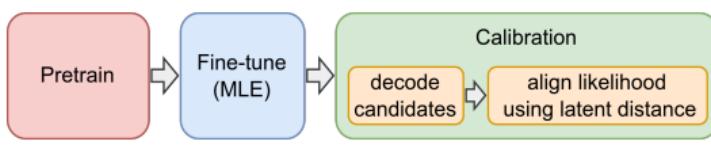
- Require annotation at the token level
- Multi-branch decoder ⇒ find the good balance (fluency, faithfulness, ...)

¹Rebuffel et al, Data Mining and Knowledge Discovery 2022.
Controlling hallucinations at word level in data-to-text generation.



Faithfulness opti as a post-processing step³

- Calibrating the likelihood in the beam-search procedure



- Conditional PMI Decoding²: detecting hazard (entropy) + shifting proba

$$\text{score}(y \mid \mathbf{y}_{<t}, \mathbf{x}) = \log p(y \mid \mathbf{y}_{<t}, \mathbf{x}) - \lambda \cdot \mathbb{1}_{\{H(p(\cdot \mid \mathbf{y}_{<t}, \mathbf{x})) \geq \tau\}} \cdot \log p(y \mid \mathbf{y}_{<t})$$

²van der Poel et al.; EMNLP 2022

Mutual Information Alleviates Hallucinations in Abstractive Summarization

³Zhao et al.; ICLR 2023

Calibrating Sequence likelihood Improves Conditional Language Generation



Let's optimize preferences ! [PPO⁴]

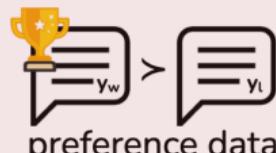
(Major) assumption

We have hallucinated *vs* proper sentences in a data-to-text framework

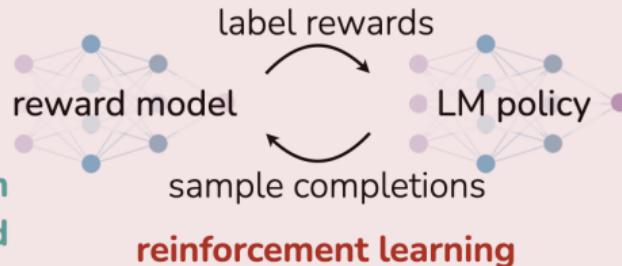
Since instructGPT... We use PPO (Proximal Policy Optimization)

Reinforcement Learning from Human Feedback (RLHF)

x: "write me a poem about
the history of jazz"



maximum
likelihood



⁴Schulman et al. arXiv 2017. Proximal Policy Optimization Algorithms

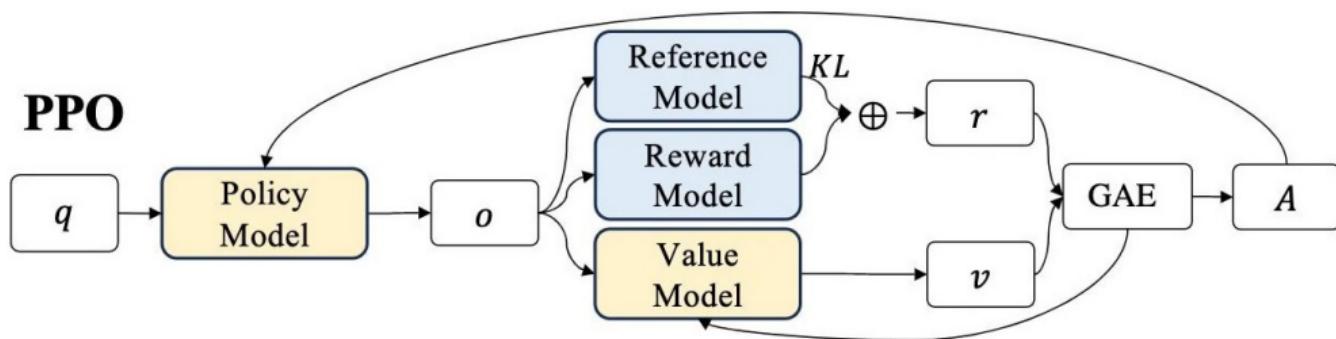


Let's optimize preferences ! [PPO⁴]

(Major) assumption

We have hallucinated *vs* proper sentences in a data-to-text framework

Since instructGPT... We use PPO (Proximal Policy Optimization)



⁴Schulman et al. arXiv 2017. Proximal Policy Optimization Algorithms

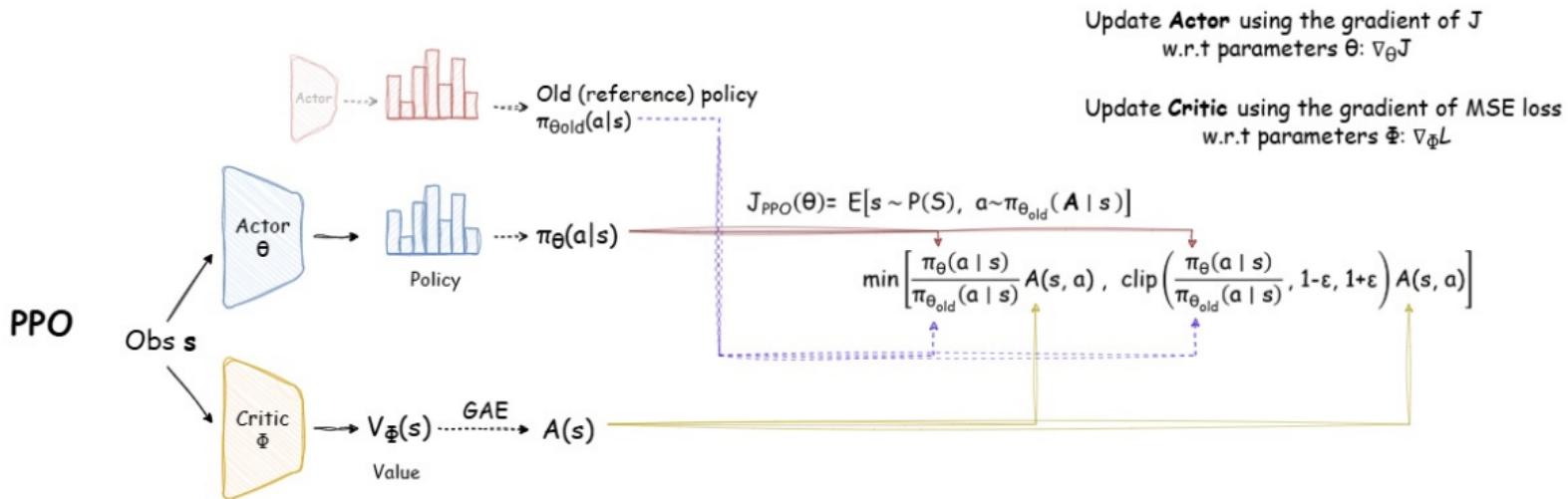


Let's optimize preferences ! [PPO⁴]

(Major) assumption

We have hallucinated vs proper sentences in a data-to-text framework

Since instructGPT... We use PPO (Proximal Policy Optimization)



⁴Schulman et al. arXiv 2017. Proximal Policy Optimization Algorithms



Let's optimize preferences ! [PPO⁴]

(Major) assumption

We have hallucinated *vs* proper sentences in a data-to-text framework

Since instructGPT... We use PPO (Proximal Policy Optimization)

- 4 models to load in memory (π_θ, π_0, V)
- 2 models with gradients ($\pi_\theta, V/A$)
- Intensive sampling $\propto \pi_\theta(y_t | y_{<t})$
- Instable procedure (cf regul. terms)

⁴Schulman et al. arXiv 2017. Proximal Policy Optimization Algorithms



Let's optimize preferences ! [PPO⁴]

(Major) assumption

We have hallucinated vs proper sentences in a data-to-text framework

Since instructGPT... We use PPO (Proximal Policy Optimization)

(Données humaines)



Comparaisons ↴

Train Reward Model r_ϕ



Prompts → LLM (π_θ) → Réponses



Eval avec r_ϕ



Recompense + KL + Avantage



PPO update $\pi_\theta \leftarrow \pi_\theta + \Delta\theta$

$$L^{\text{CLIP}}(\theta) = -\mathbb{E} \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_t \right) \right]$$

Reward:

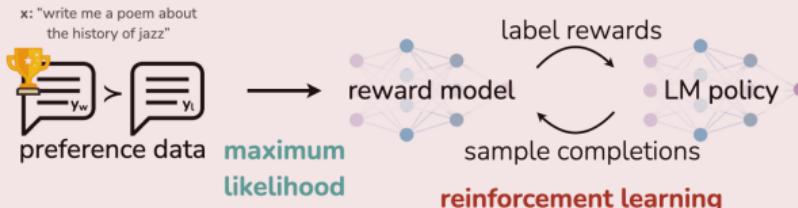
$$r_t(\theta) = \beta \log \frac{\pi_\theta(y_t | y_{<t})}{\pi_0(y_t | y_{<t})}$$

⁴Schulman et al. arXiv 2017. Proximal Policy Optimization Algorithms

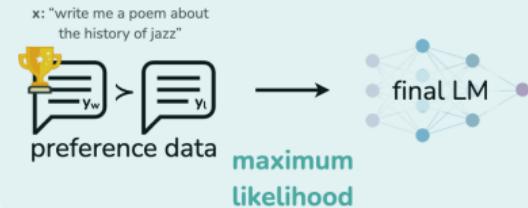


Simplifying the procedure [DPO⁵]

Reinforcement Learning from Human Feedback (RLHF)



Direct Preference Optimization (DPO)



Same reward:

$$\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y_t | y_{<t})}{\pi_0(y_t | y_{<t})}$$

Different cost:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(y_{t+}, y_{t-}, y_{<t}) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_{t+} | y_{<t})}{\pi_0(y_{t+} | y_{<t})} - \beta \log \frac{\pi_\theta(y_{t-} | y_{<t})}{\pi_0(y_{t-} | y_{<t})} \right) \right]$$

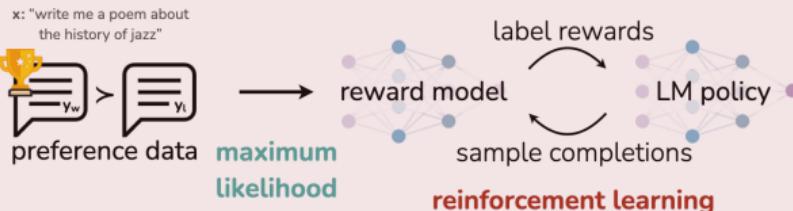
⁵Rafailov et al., NeurIPS 2023.

Direct Preference Optimization: Your Language Model is Secretly a Reward Model



Simplifying the procedure [DPO⁵]

Reinforcement Learning from Human Feedback (RLHF)



Direct Preference Optimization (DPO)



- 2 models to load in memory (π_θ, π_0)
- 1 models with gradients (π_θ)
- Intensive sampling but $\propto \pi_0(y_t | y_{<t})$ ⇒ enable precomputing
- Classical (=stable) likelihood optimization

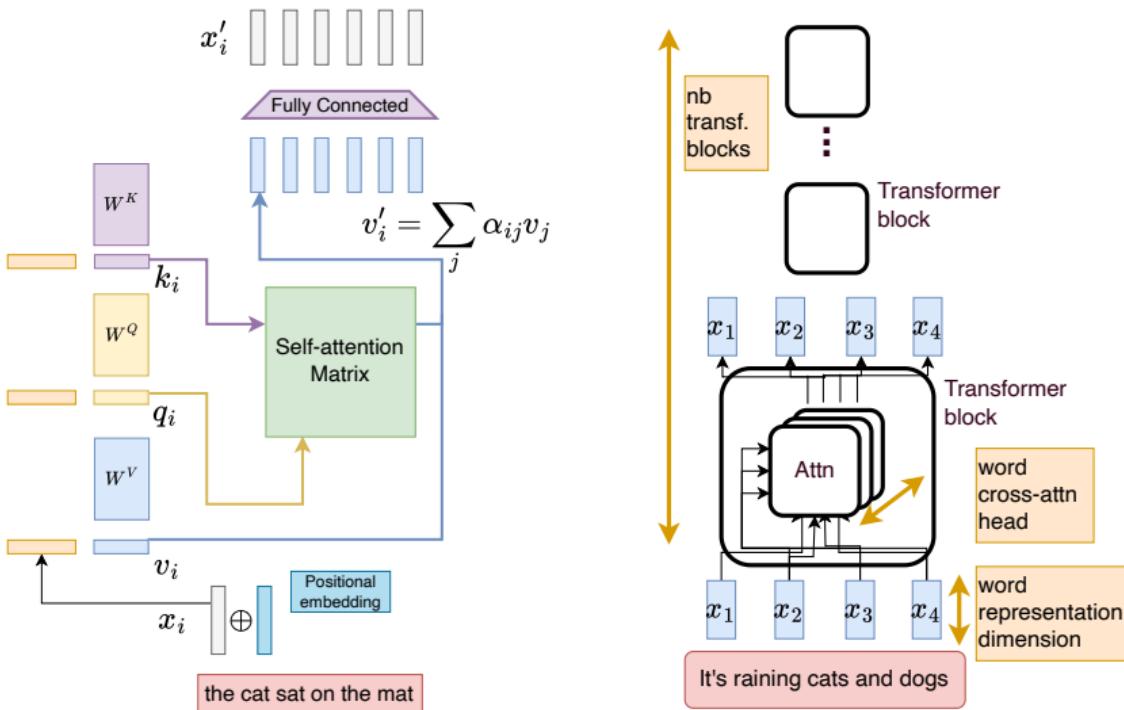
⁵Rafailov et al., NeurIPS 2023.

BACK TO THE ENCODERS



BERT : 2018 milestone

■ Efficient transformer architecture



Several sizes

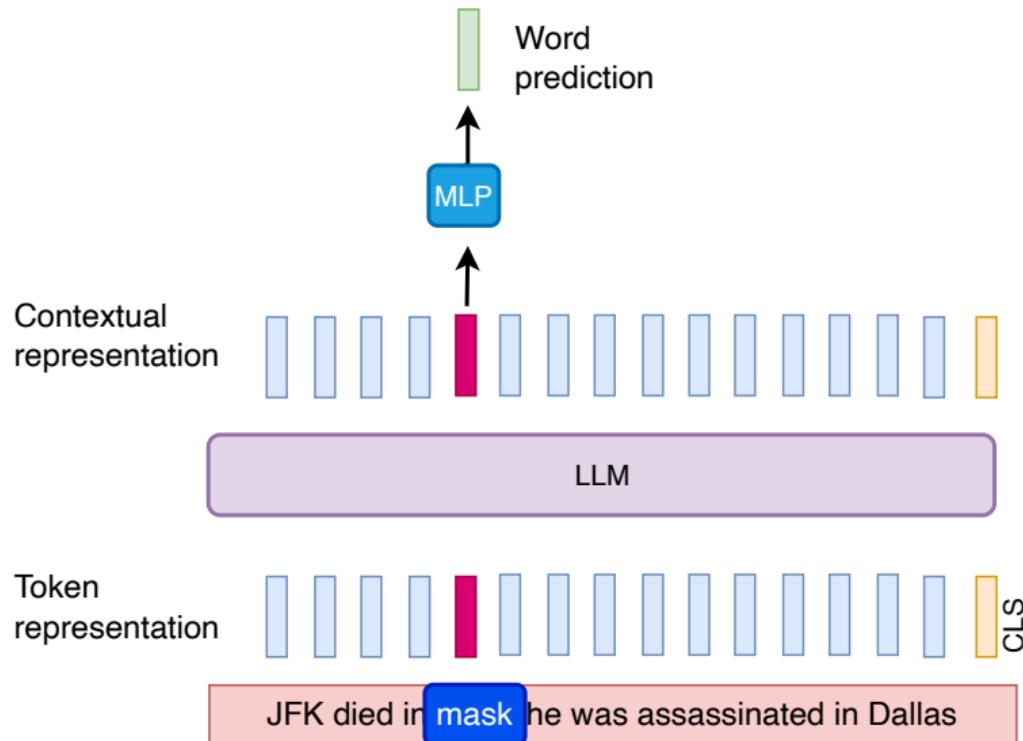
Nice implementation / pretraining

HuggingFace Integration



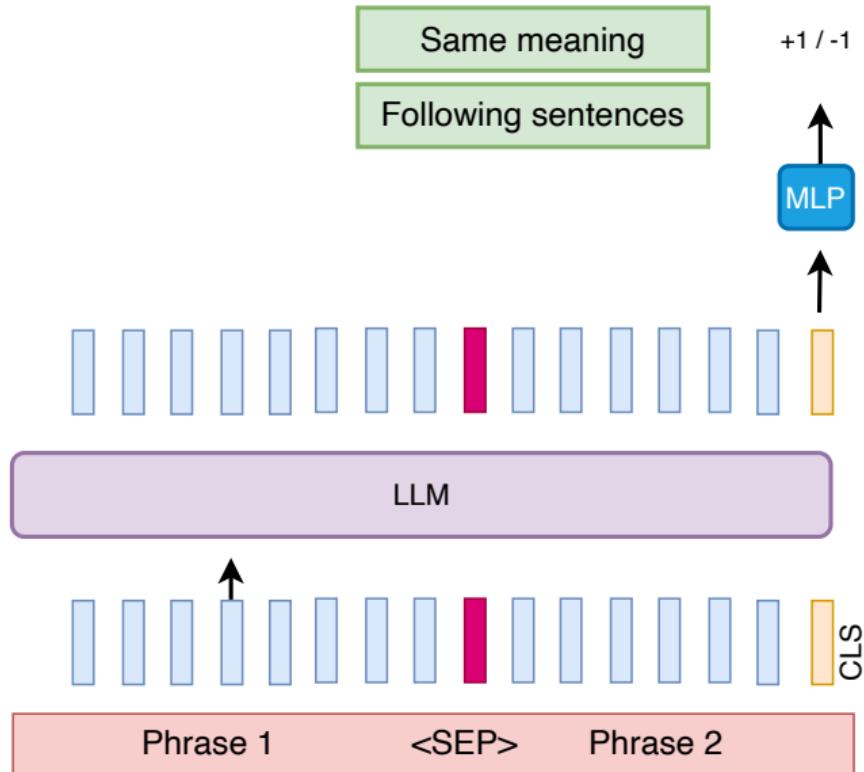
BERT Pre-training

■ Masking words



BERT Pre-training

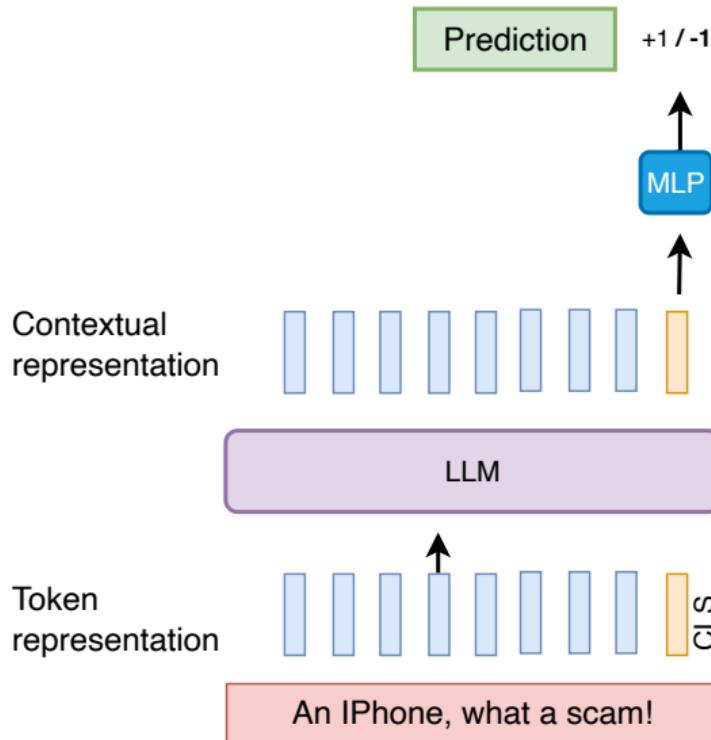
- Predicting coherent / successive sentences





Then tackling various tasks

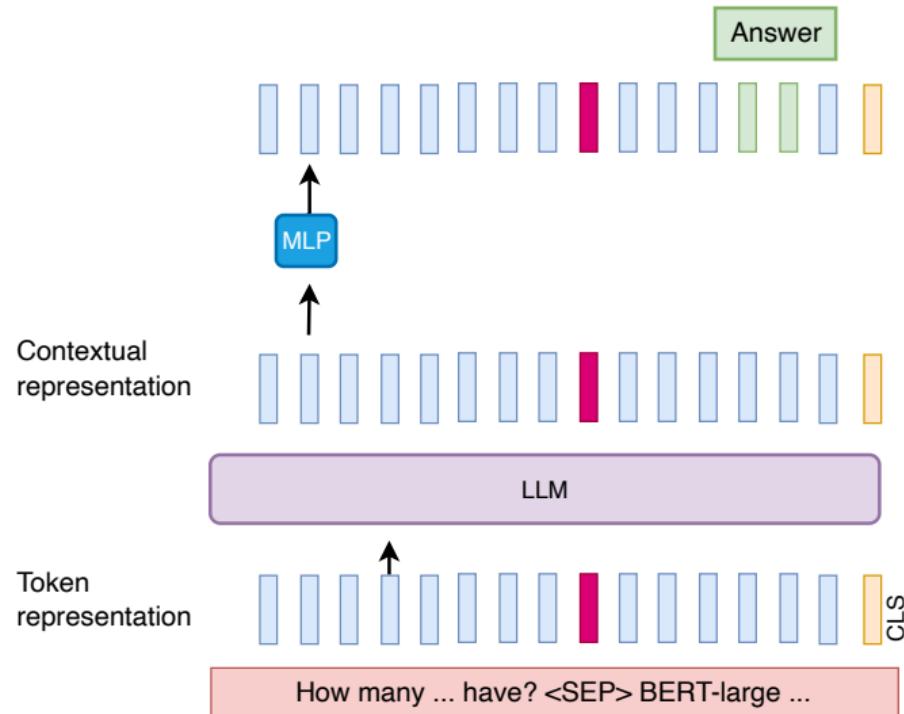
■ Opinion mining





Then tackling various tasks

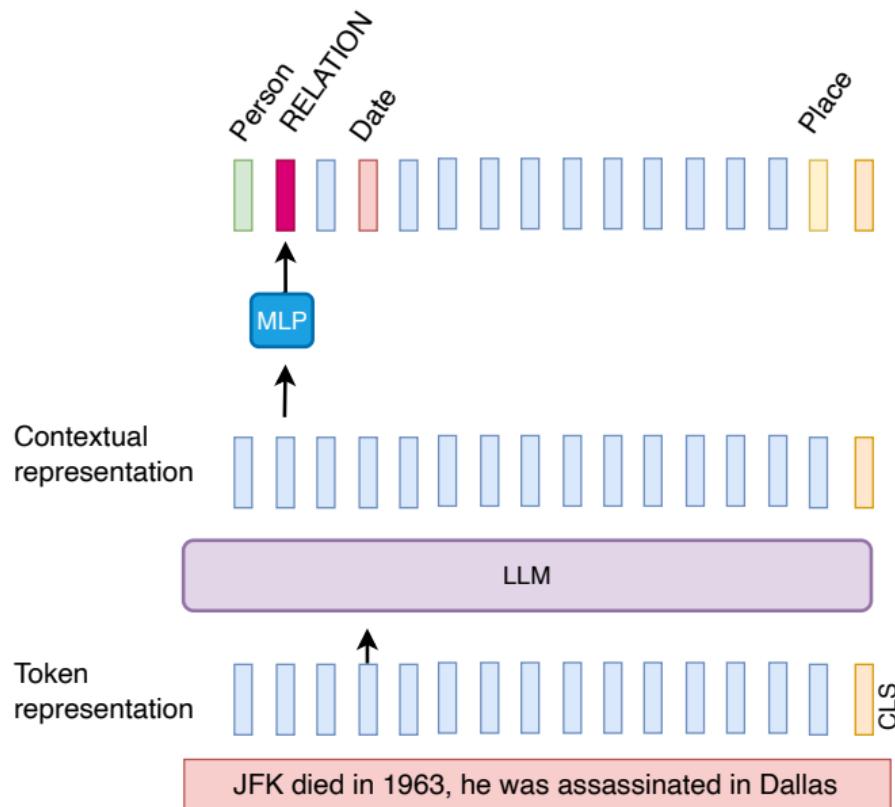
- Question Answering
 - Often couple with Info Retrieval
- ⇒ finding short passages to analyse





Then tackling various tasks

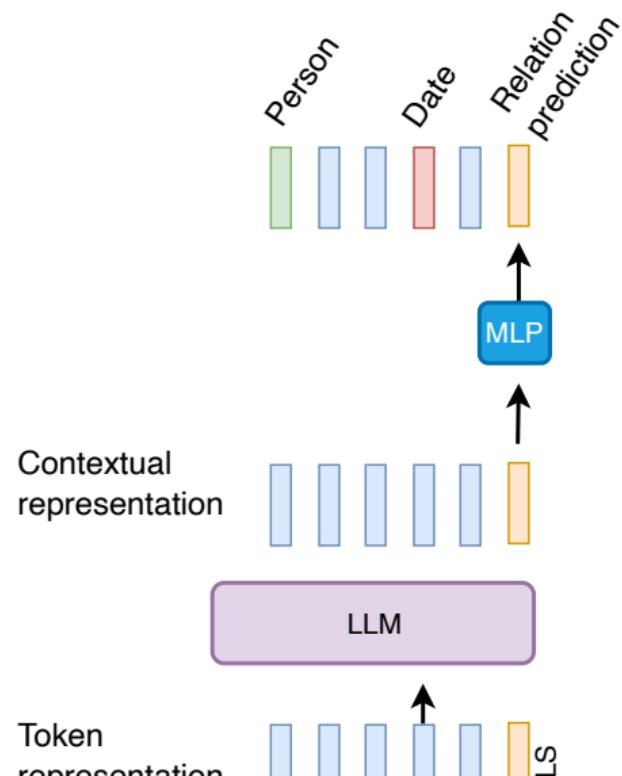
- Information extraction
- Named Entity recognition
- Relation extraction
- ... but on which word ??





Then tackling various tasks

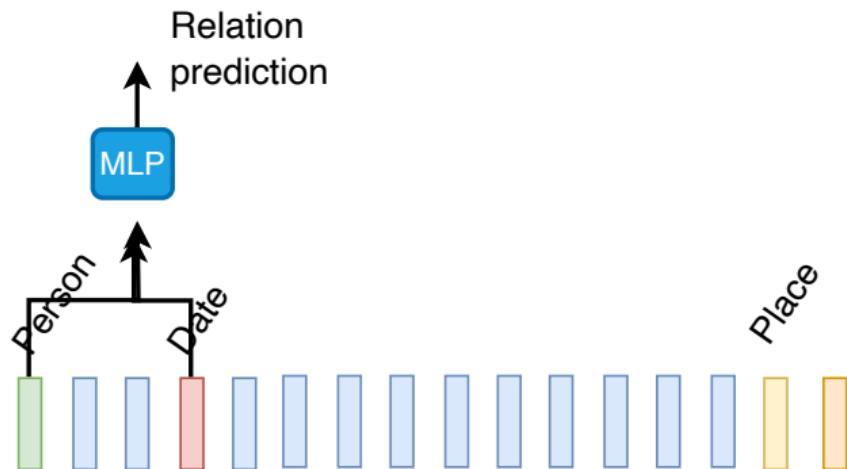
- Information extraction
- Named Entity recognition
- Relation extraction
... but on which word ??
- Hyp : unique relation in the sentence





Then tackling various tasks

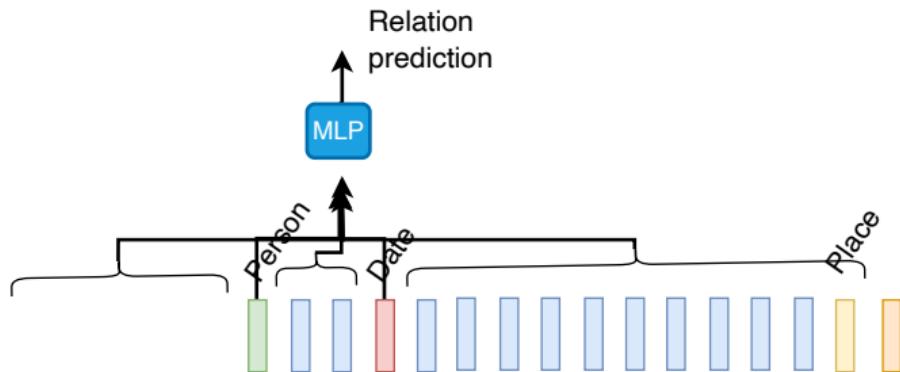
- Information extraction
- Named Entity recognition
- Relation extraction
 - ... but on which word ??
- Hyp : unique relation in the sentence
- Between entities + all couple tested





Then tackling various tasks

- Information extraction
- Named Entity recognition
- Relation extraction
 - ... but on which word ??
- Hyp : unique relation in the sentence
- Between entities + all couple tested
- Piecewise representation (PCNN)



EFFICIENT TUNING

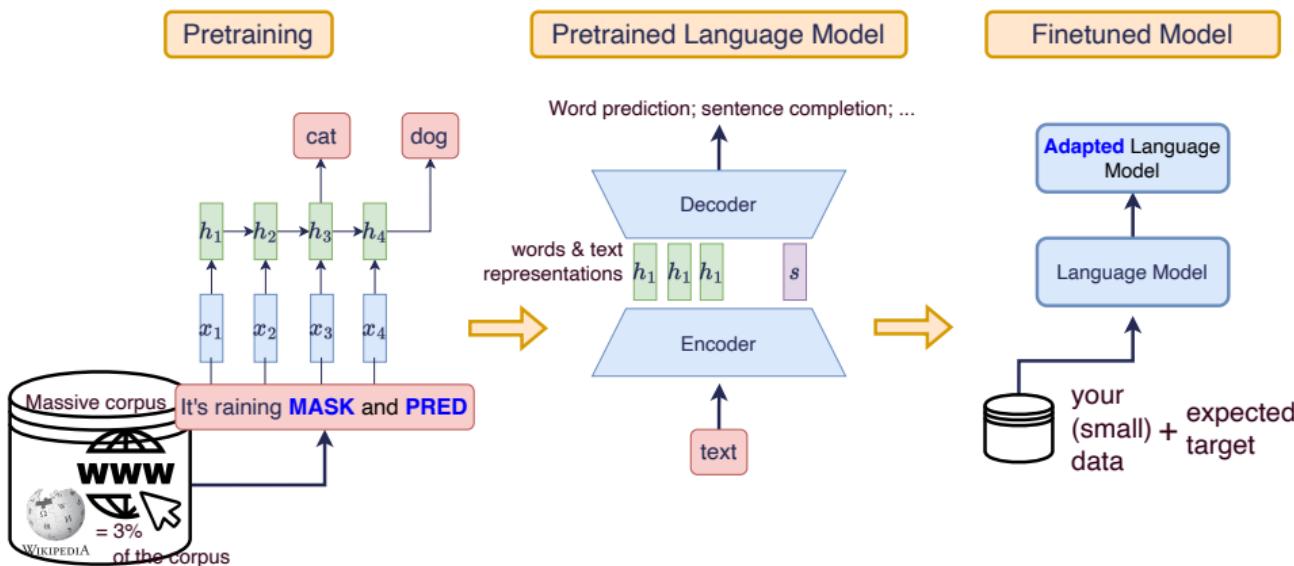
BY
JONATHAN H. LEE



Once the pretraining done

How to update/upgrade a model for a specific domain/task?

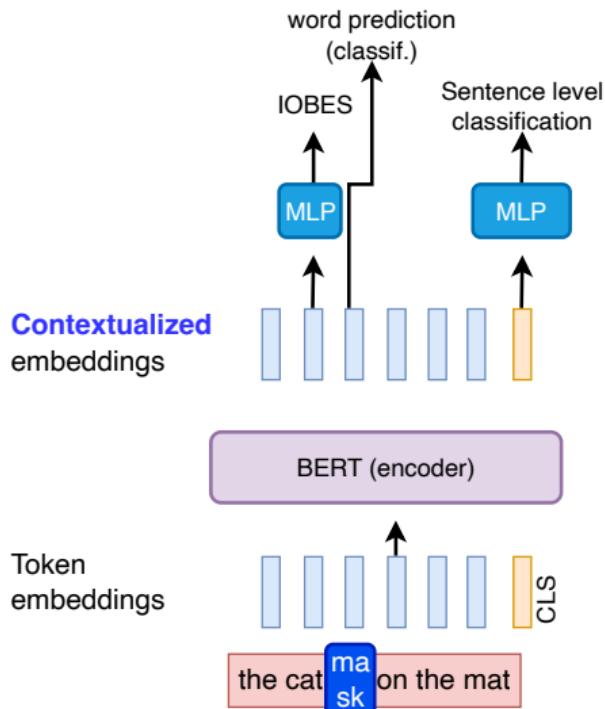
- ⇒ (Relatively) cheap update
- ⇒ Few data required





Refining the encoder

- Assuming we have textual data & supervision \Rightarrow optimizing the models



- Supervision: X : texts; Y : targets;
 $(+ W$: parameters)
- Finetuning: refining W s.t. $f_W(X) \approx Y$

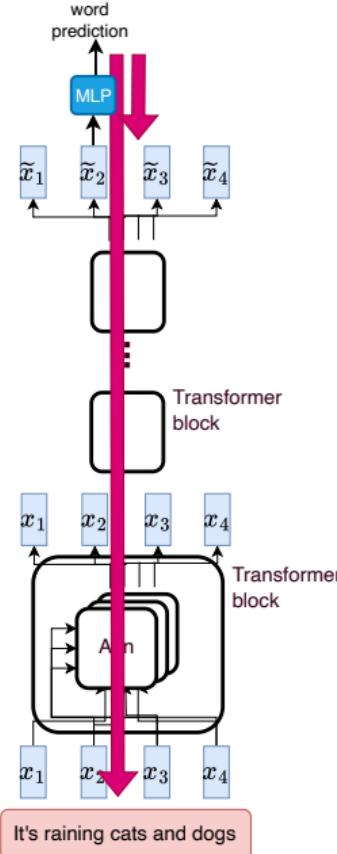
May a **small** model become more efficient than a **big** LLM through refinement?

How to make the refinement process cheap?



Efficient Tuning

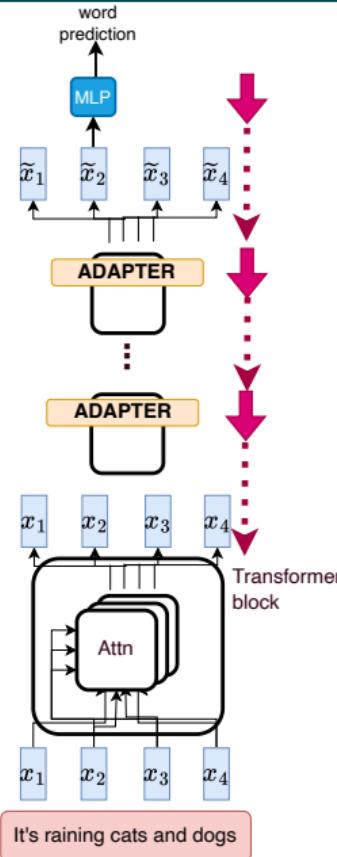
- Limiting the number of epoch
 - Running on few data
- Only optimize the last layer





Efficient Tuning: main strategies

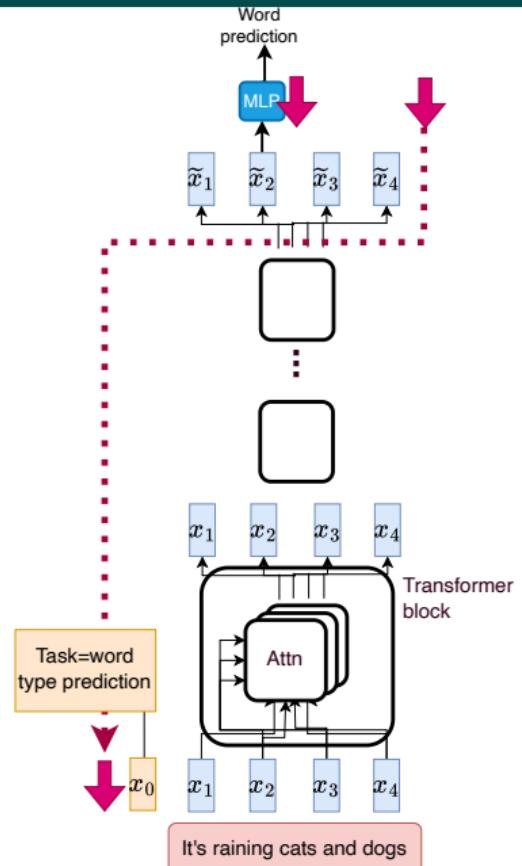
- 1 Adapter
- 2 Prompt Tuning
- 3 Low Rank Adaptation





Efficient Tuning: main strategies

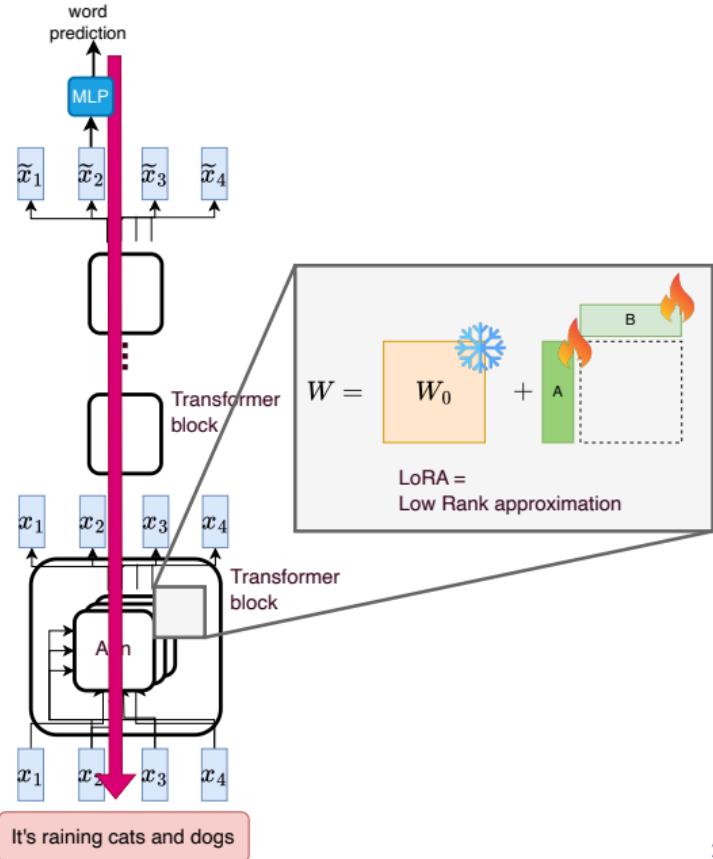
- 1 Adapter
- 2 Prompt Tuning
- 3 Low Rank Adaptation





Efficient Tuning: main strategies

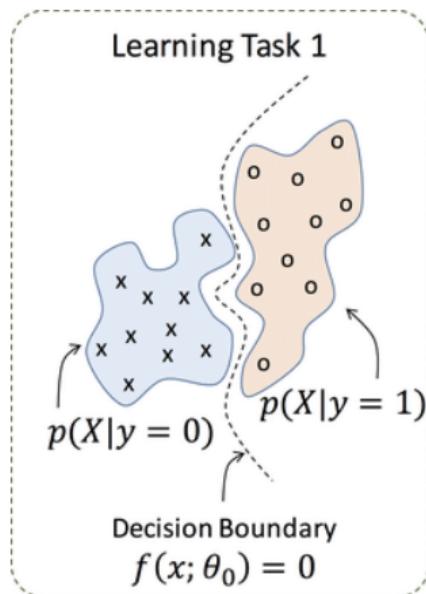
- 1 Adapter
- 2 Prompt Tuning
- 3 Low Rank Adaptation



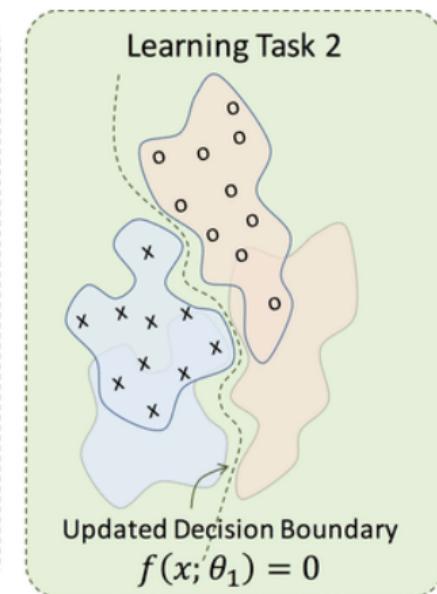


Catastrophic forgetting

Learning new skills... Forgetting the bases?



Change
in $p(X)$



CONCLUSION



Generative AI : new opportunities at a cost

- **LLM cost** >> encoders (>> CRF / trad. ML >> classical lexical rules)
 - Paying for extraction *or* for development?
- New formulations ⇒ new metrics ⇒ **how to evaluate properly?**
- Above which threshold the systems are **useful**? Autonomously useful?
- What about a system that would be able to extract *statements*?

Transformer le texte suivant en une liste de statements élémentaires:

Des drones non identifiés ont été observés au-dessus de la plus grande base militaire du Danemark vendredi soir, a annoncé la police, après plusieurs survols d'aéroports cette semaine dans le pays nordique.

Voici une décomposition en statements élémentaires :

1. Des drones non identifiés ont été observés.
2. L'observation a eu lieu au-dessus de la plus grande base militaire du Danemark.
3. L'observation a eu lieu vendredi soir.
4. La police a annoncé cette observation.
5. Des survols d'aéroports ont eu lieu cette semaine.
6. Les survols d'aéroports ont eu lieu dans le Danemark.
7. Le Danemark est un pays nordique.