

NLP: TRANSFORMER

Agro-IODAA-Semestre 1



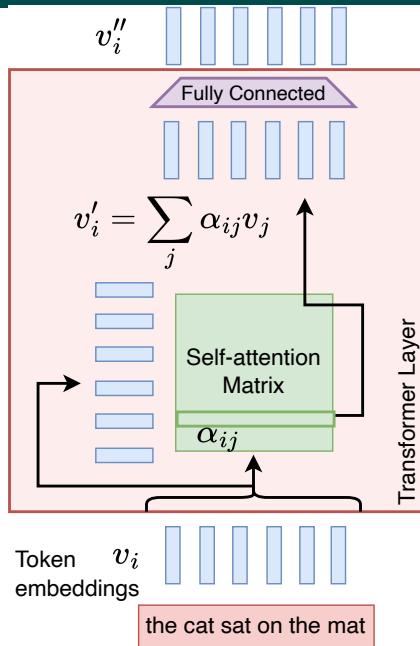
Vincent Guigue





Architecture

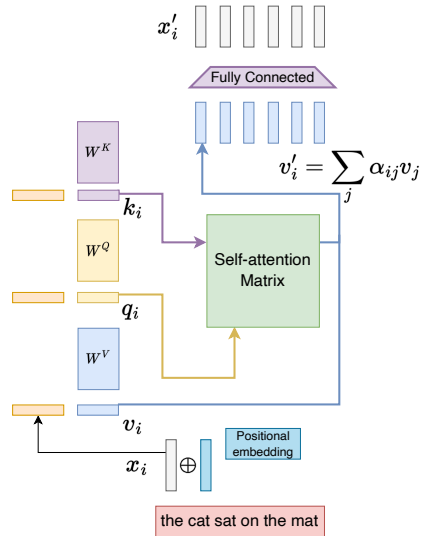
- Identifier les **entrées** et les **sorties**
- Différentier la partie *transformer* et la partie *décision*





Architecture

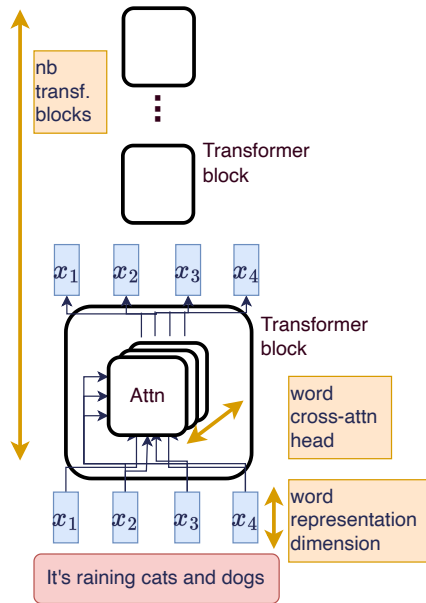
- Identifier les **entrées** et les **sorties**
- Différencier la partie *transformer* et la partie *décision*





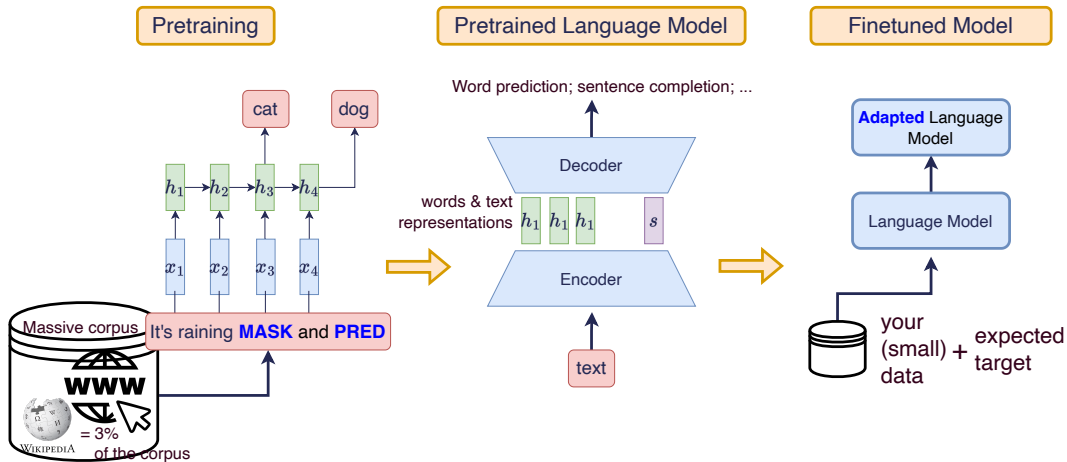
Architecture

- Identifier les **entrées** et les **sorties**
- Différentier la partie *transformer* et la partie *décision*





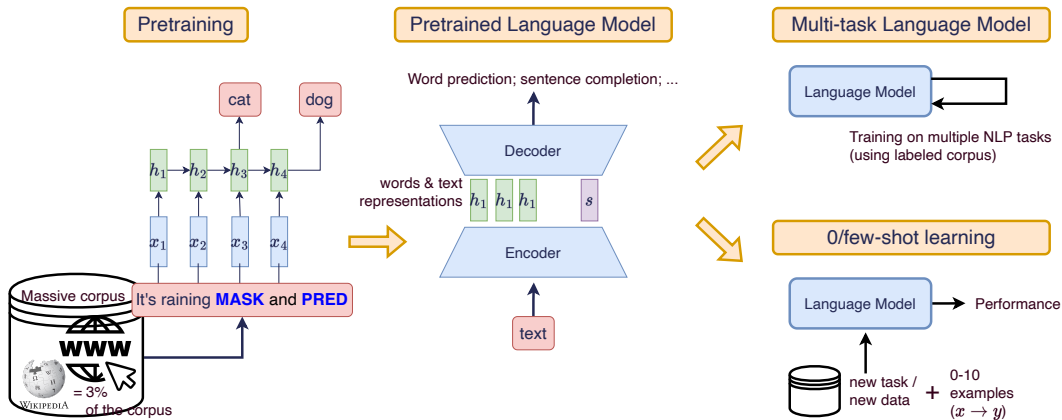
Paradigme: re-utilisation



■ Qu'est ce qu'une *adaptation* de modèle?



Paradigme: re-utilisation





Chaine de traitements

- 1 Tokenisation
- 2 Transformer
- 3 Décision

Clés de compréhension

- Dépendances des traitements
- Compréhension des branchements
- Quid de la langue? (Français / Anglais / ...)
- Taille des entrées
 - Padding
 - Truncation



- Acteur incontournable en texte:
 - Généricité des architectures
 - Disponibilité des modèles: un hub qui s'entretient
 - ... Une ouverture vers d'autres modalités



`https://huggingface.co`

- Qu'est ce qu'un *foundation model*?
- Pourquoi est ce que ça semble important pour le futur?



Défis :

D1: Analyser la représentation des pronoms

Proposer un protocole pour étudier les variations de représentation des pronoms hors contexte vs dans le contexte

- Représentation des stop-words: word2vec vs LLM
- Déambiguïsation des termes polysémiques

D2: Clustering de textes

- Construire un algorithme de résumé automatique sur la base d'une méthode de clustering



D3: Construire/récupérer un classifieur d'opinion

- Comparer les performances avec le modèle sac de mots
- Envoyer des décisions sur les exemples de test avec les deux approches
- Réfléchir sur la validité des résultats proposés

D4: Construire un classifieur d'auteur

- Sur les données Chirac/Mitterrand
- Comparer les performances avec le modèle sac de mots
- Envoyer des décisions sur les exemples de test avec les deux approches
- Réfléchir sur la validité des résultats proposés
- Quid de la langue



Pourquoi décortiquer un modèle?

- Adapter la/les fonction(s) de coût
 - Brancher plusieurs modules de décision
- ⇒ Savoir le construire et savoir l'entraîner

Une nouvelle série de questions sur les composants:

- Qu'est ce qu'un DataLoader ?
- Qu'est ce qu'un Optimizer/Scheduler ?



Chaine d'apprentissage standard

```
1  for epoch in range(num_epochs):
2      for batch in train_dataloader:
3          batch = {k: v.to(device) for k, v in batch.items()}
4          outputs = model(**batch)
5          loss = outputs.loss
6          loss.backward()
7
8          optimizer.step()
9          lr_scheduler.step()
10         optimizer.zero_grad()
11         progress_bar.update(1)
```

Supervision distante / réduction de modèle

- Quel impact sur les performances?
- Architecture itérative... Quels risques?
- Mécanisme de distillation

