Soutenance Stage L3

Machine-learning models to study how children develop in their ability to take turns in conversations

Martin Cuingnet et Abdellah Fourtassi, équipe TALEP du LIS

2024-11-02

ENS Paris-Saclay

1. Contexte et objectifs

Contexte

Décomposition en différents tours de paroles

Alice parle puis Bob puis Alice, etc...

Plusieurs canaux utilisés pendant un tour :

- Le **Main Channel** : élément principal d'un tour de parole
- Le **Back Channel** : intervention mineure, verbale ou non-verbale

Contexte

Alice : Salut !

Bob : Salut !

Alice : Tu ne sais pas la dernière ?

Bob : Non ?

Alice : Aujourd'hui le facteur n'est pas passé !

Bob : Non !

Alice : On m'a même raconté qu'il ne passera jamais ! Quelle histoire !

Bob : [Hochement de tête]

Alice : Et moi qui attendais du courrier... [Lève les yeux au ciel]

L'alternance des couleurs représente la succession des tours de paroles avec en **gras le Main Channel** et en *italique le Back Channel*.

Prédiction des tours de paroles

Alice : salut

Bob : salut

Alice : tu ne sais

On ne prédit pas la fin du tour de parole

Alice : salut

Bob : salut

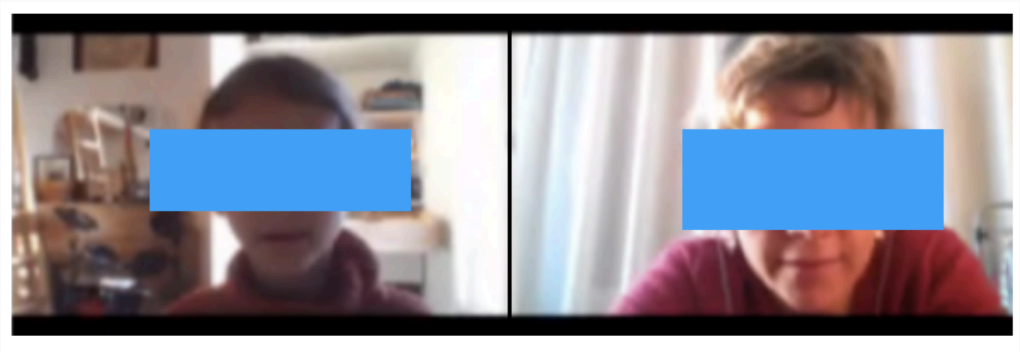
Alice : tu ne sais pas la dernière

On prédit la fin du tour de parole

Dataset

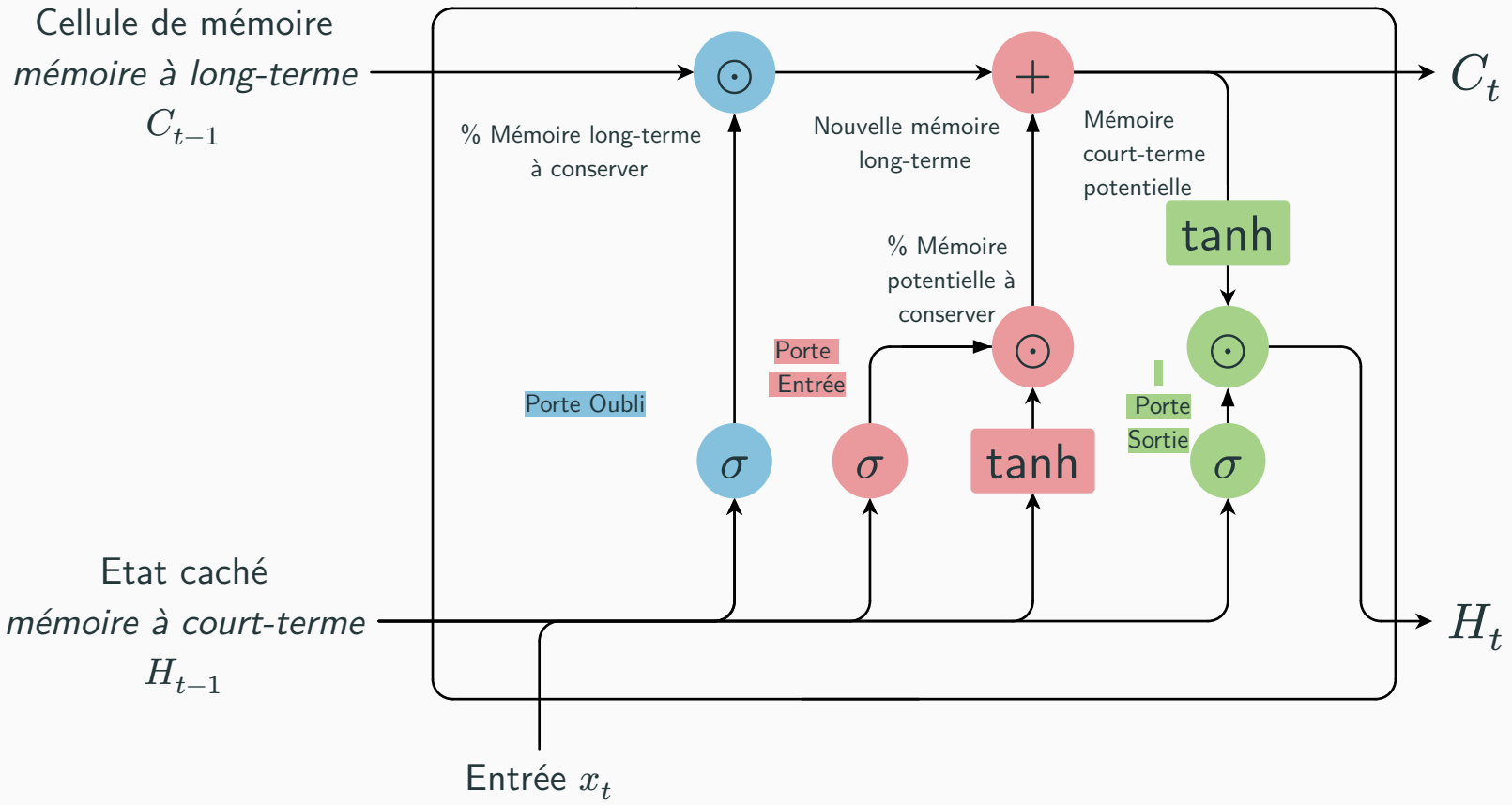
On réalise cette tâche de prédiction sur le jeu de donnée **ChiCo** :

- 40 conversations Zoom d'environ 10 minutes entre enfant et parent et entre parent et un autre adulte
- Features audio (son de la voix), visuelles (sourire) et verbales (classe grammaticale du mot prononcé) extraites au préalable



2. Première approche : les LSTM

LSTM *(Compréhension du modèle : 3 jours)*



- Long Short-Term Memory
- Modèle Récurrent
- Appliqué autant de fois que nécessaire pour couvrir l'intervalle de 2 secondes

Base de travail : **Development of Multimodal Turn Coordination in Conversations: Evidence for Adult-like behavior in Middle Childhood** par Abdellah Fourtassi et all

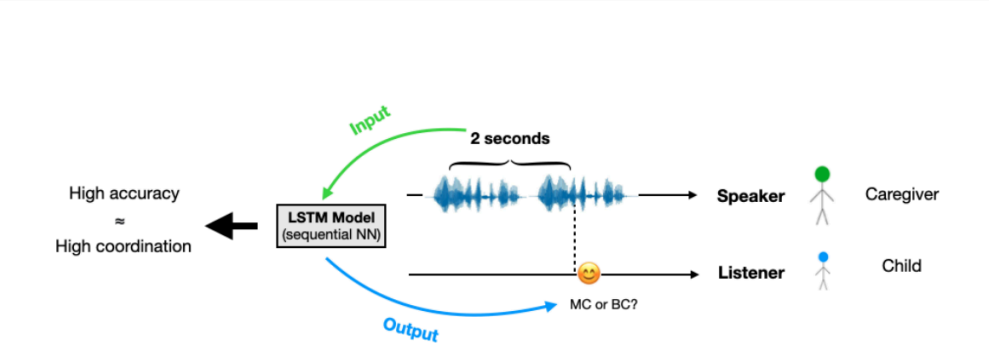


Fig. 1. – Procédure expérimentale

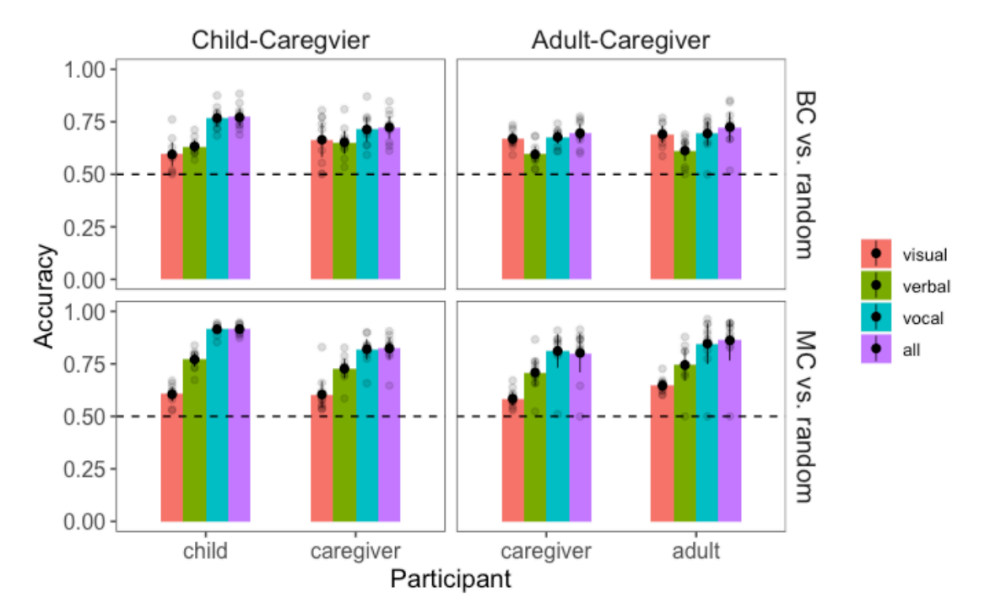


Fig. 2. – Résultats

3. Transformer

Nouvelle perspective : les Transformers

Après avoir reproduit les résultats de mon encadrant avec LSTM *(3 jours)* : nouvelle approche avec l'**architecture des transformers**

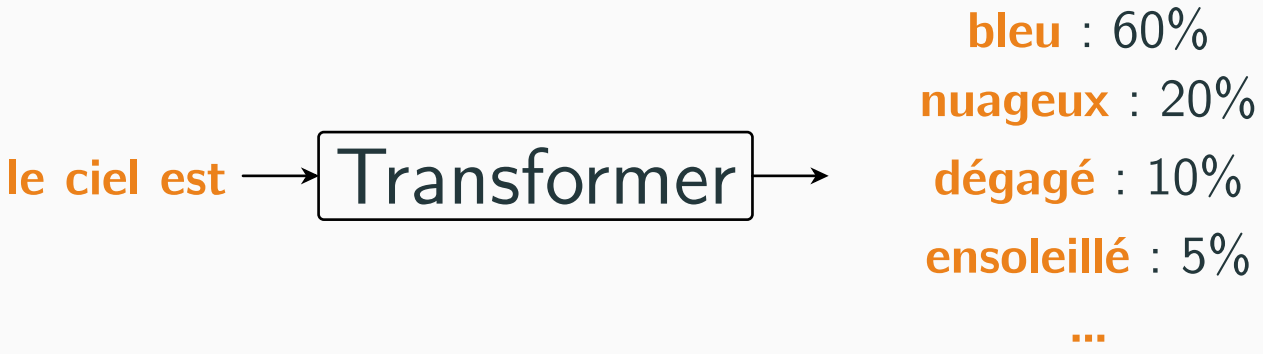
Raisons :

- Sous exploitation de la modalité verbale : « je pense à un animal » → « *pronom verbe conjonction déterminant nom* »
- Non parallélisable : restreint à un faible nombre de paramètre comparé aux transformers
- Modèle dépassé dans le domaine du TAL

Objectif du transformer *(Compréhension du modèle : 3 jours)*

Modèle introduit dans le papier **Attention is All You Need** : dans le cas de ce travail, modèle auto-attentif

Modèle permettant de faire de la prédiction de texte



On va ici l'utiliser pour prédire les changements de tour

Première étape : la tokenization

Le modèle ne manipule pas directement les mots de la phrases mais des **tokens**

the sky is blue → [1820, 13180, 374, 6437]

le ciel est bleu → [273, 12088, 301, 1826, 12704, 84]

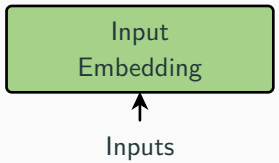
Avant d'être passé au modèle, le texte est converti en une suite de tokens, et le modèle essaiera de prédire le prochain

Plongement lexical

A chaque token, on associe un vecteur qui représente sa sémantique : son **embedding**
(de 12288 dimensions dans le cas de GPT-2)

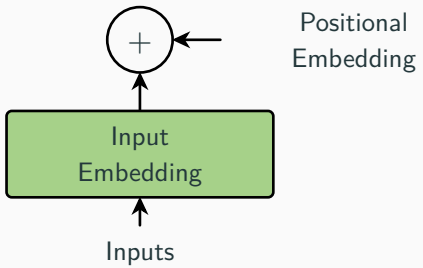
On aura par exemple :

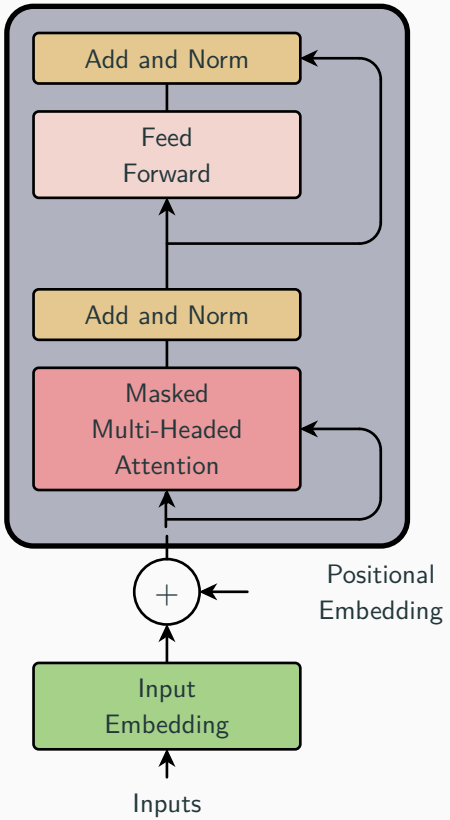
$$\overrightarrow{\text{woman}} - \overrightarrow{\text{man}} + \overrightarrow{\text{king}} \simeq \overrightarrow{\text{queen}}$$



Embedding positionnel

Selon la position du token dans la séquence, on lui ajoute un embedding positionnel : un vecteur qui représente de manière unique sa position



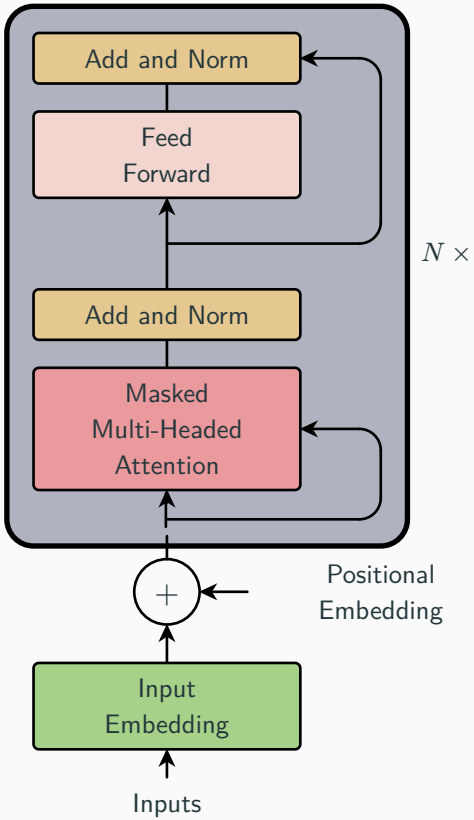


Mécanisme d'attention

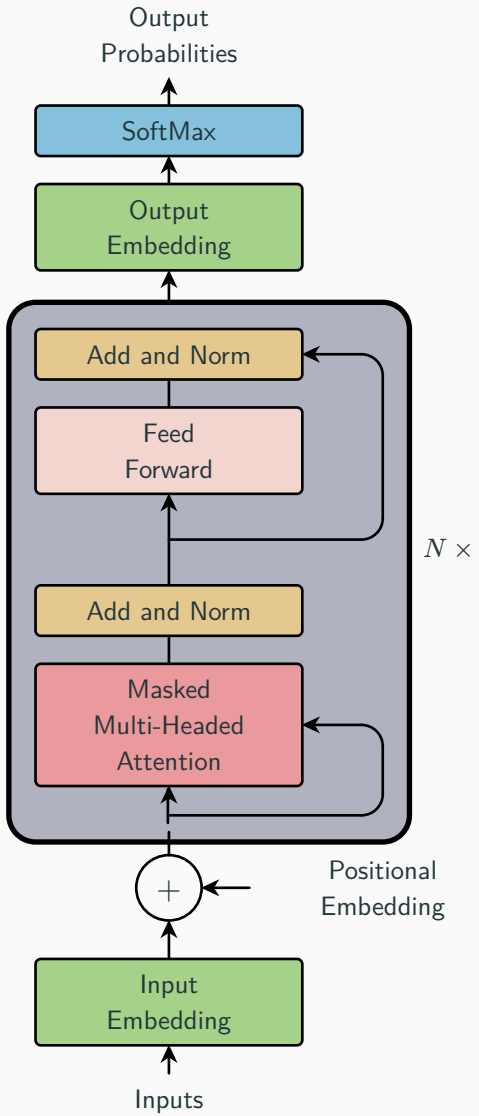
Permet aux différents tokens de « communiquer » entre eux et de s'imprégner de la sémantique des autres tokens de la séquence au travers de « questions » et de « réponses »

Un token pourrait se demander si il est bleu en demandant aux autres tokens si ils se réfèrent à cette couleur
(Voir explication plus détaillée en annexe)

Explication des transformers auto-attentifs



On répète ensuite le mécanisme d'attention et de feed forward pour chaque couche du transformer (avec des poids différents pour chaque couche)



Plongement lexical inverse

A la fin du processus, la sémantique du mot à prédire est encapsulée dans le vecteur associé au dernier token.

On applique alors une transformation inverse à la matrice d'embedding pour obtenir le score du vecteur pour chaque token (plus le vecteur a un score élevé pour un token, plus sa sémantique se rapproche de ce dernier)

4. Formalisation et entraînement

- LSTM : prédiction direct du channel
- Transformer : prédit prochain mot (token) d'une conversation

⇒ **Adapter le problème**

- Restreindre modalité textuelle
- Introduire token spécial changement de tour : `<|turnshift|>` : Ajout d'une colonne dans la matrice d'embbeding

Exemple de conversation après mise en forme

---

salut <|turnshift|> salut <|turnshift|> tu ne sais pas la dernière <|turnshift|> non <|turnshift|> aujourd'hui le facteur n'est pas passé <|turnshift|> non <|turnshift|> on m'a même raconté qu'il ne passera jamais quelle histoire et moi qui attendais du courrier

---

L'alternance des couleurs représente la succession des tours de paroles

---

Entraînement

Modèle pré-entraîné finetuné sur le transcript de ChiCo mis en forme (modèle entraîné pour prédire le prochain mot quand on lui donne une conversation dans le style du transcript)

Évaluation

Pour une conversation du transcript, en notant $n = \#\{<|\text{turnshift}|> \in \text{conversation}\}$

- n tests où il faut prédire `<|turnshift|>`
- n tests où il ne faut pas prédire `<|turnshift|>`

Exemple : on donne au modèle **salut <|turnshift|> salut <|turnshift|> tu ne sais pas la** et il doit prédire un token différent de `<|turnshift|>`

5. Expériences ChiCo

Experiences ChiCo *(1 semaine)*

1. **gpt2-large** sur le transcript français
2. **gpt2-large** sur le transcript traduit en anglais
3. **claire-7B** sur le transcript français en utilisant uniquement du prompting

Résultats ChiCo

	taille du contexte	**75 tokens**
langue du dataset	**threshold** / **modèle**	5 %
FR	gpt2-large (774M)	84.2 %
EN	gpt2-large (774M)	88.0%
FR	Claire-7B (7B)	79%

Tableau 1. – Taux de réussite pour les différents modèles pour ChiCo

6. ChiCA

Le dataset ChiCa

Après création du dataset ChiCo, Fourtassi et d'autres membres de l'équipe TALEP : création du dataset **ChiCa**.

- 22 conversations en face à face entre enfant et parent d'environ 10 minutes
- Enregistrées au domicile de l'enfant
- Enfant et parent munis de lunette traquant le regard

Modalité textuelle **et** visuelle

Extraction des features

Features extraites des données brutes :

- Le transcript des conversation avec des timecodes
- La position du regard sur la vidéo enregistrée
- La position du visage sur la vidéo enregistrée

Après traitement des features :

Transcript annoté, pour chaque mot : le regard est-il porté sur l'interlocuteur ?

7. Formalisation et modification de l'architecture

Formalisation *(3 jours)*

A partir du transcript annoté ⇒ séquence de tokens annotés par information binaire

On transforme ensuite le couple (`token`, bit) en un unique token :

- Pas de regard sur l'interlocuteur : `token`
- Regard sur l'interlocuteur : `~token~`

Formalisation *(3 jours)*

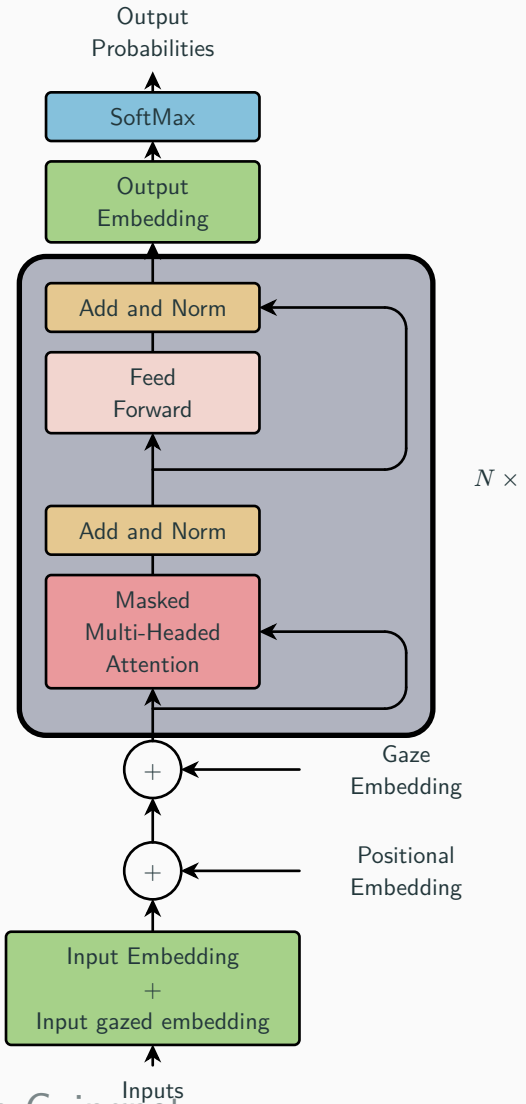
Exemple de conversation après mise en forme :

```
~salut~
<|turnshift|>
~salut~
<|turnshift|>
tu ne sais pas ~la~ ~dernière~
<|turnshift|>
~non~
<|turnshift|>
aujourd'hui le facteur n'est pas passé on m'a même raconté qu'il ne passera jamais
non mais vraiment c'est honteux et moi qui ~attendais~ ~du~ ~courier~
```

Modification de l'architecture (Conception + Implémentation : 1 semaine)

- Rajout dans la matrice d'embedding de tout les tokens de la forme `~token~` avec `token` dans la matrice d'embedding
- Pour chaque, `~token~`, on initialise son vecteur d'embedding à celui de `token` au début de l'entraînement
- On ajoute un nouveau type d'embedding à la manière du positional embedding : **le gaze embedding**

Modification de l'architecture (Conception + Implémentation : 1 semaine)



Vecteur associé au token `token` avant le mécanisme d'attention :

$$\overrightarrow{\text{token}} = \text{Input}[\text{token}_{\text{id}}] + \overrightarrow{\text{positional\_embed}}[\text{position}_{\text{token}}]$$

$$\overrightarrow{\text{token\_gazed}} = \text{Input}[\text{token\_gazed}_{\text{id}}] + \overrightarrow{\text{positional\_embed}}[\text{position}_{\text{token\_gazed}}] + \overrightarrow{\text{gaze\_vector}}$$

Avec $\overrightarrow{\text{gaze\_vector}}$ un paramètre du modèle

8. Expériences : ChiCa

Résultats *(2 jours)*

	taille du contexte	**75 tokens**
langue du dataset	**threshold** / **modèle**	**10 %**
FR	gpt2 (117M)	86.3 %
FR	gpt2-large (774M)	88.1%

Tableau 2. – Taux de réussite de différents modèles sur ChiCa avec gaze

9. Interprétabilité

Importance du contexte *(1 jour)*

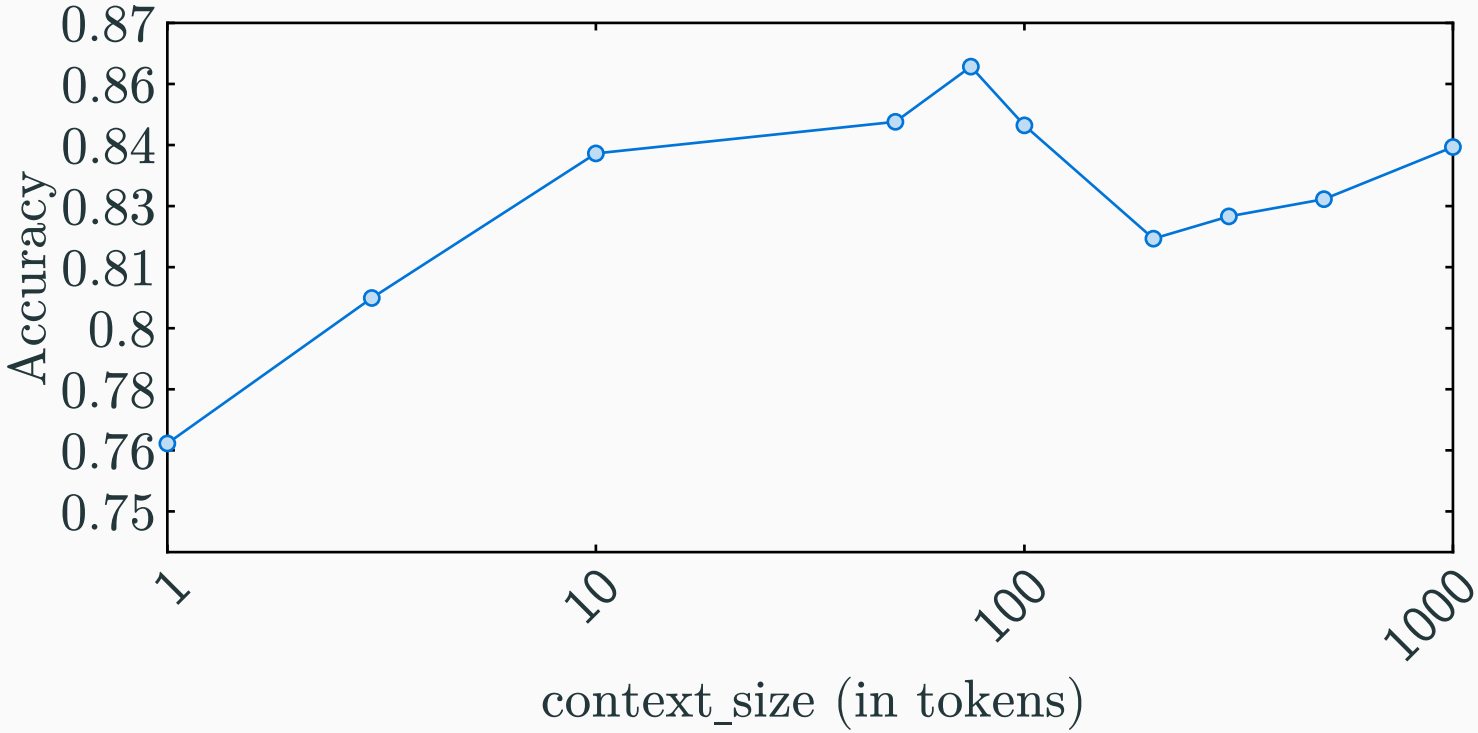


Fig. 3. – Réussite du modèle basé sur GPT2-large en fonction de la taille du contexte

Importance du contexte *(1 jour)*

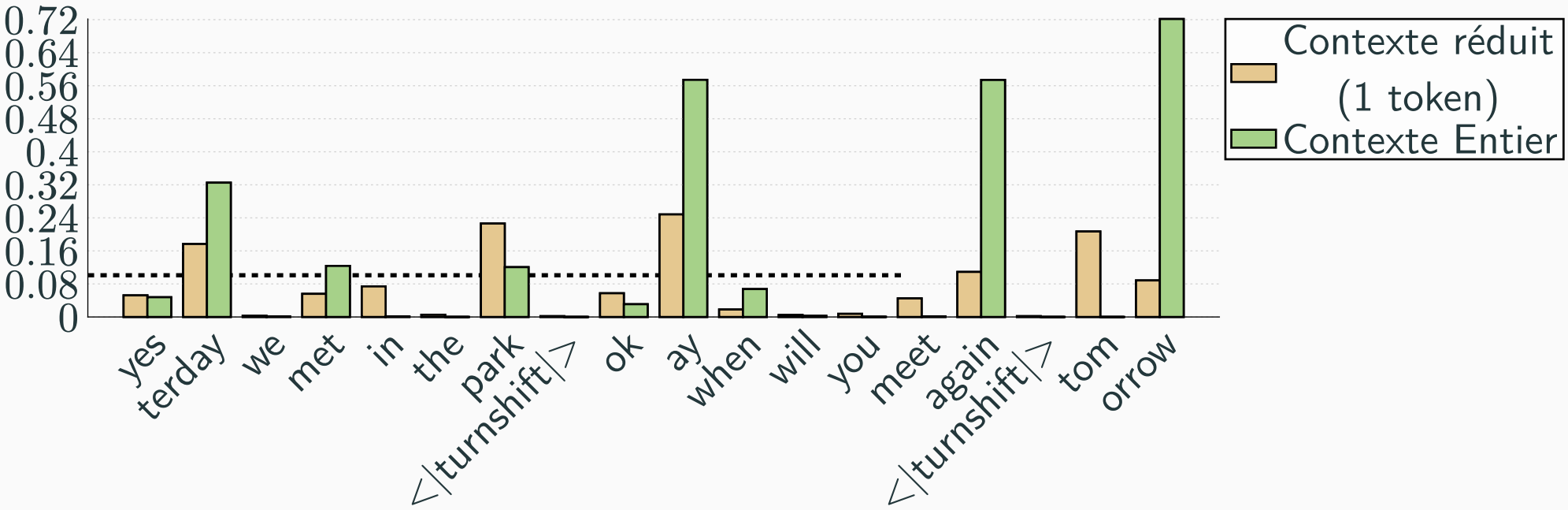
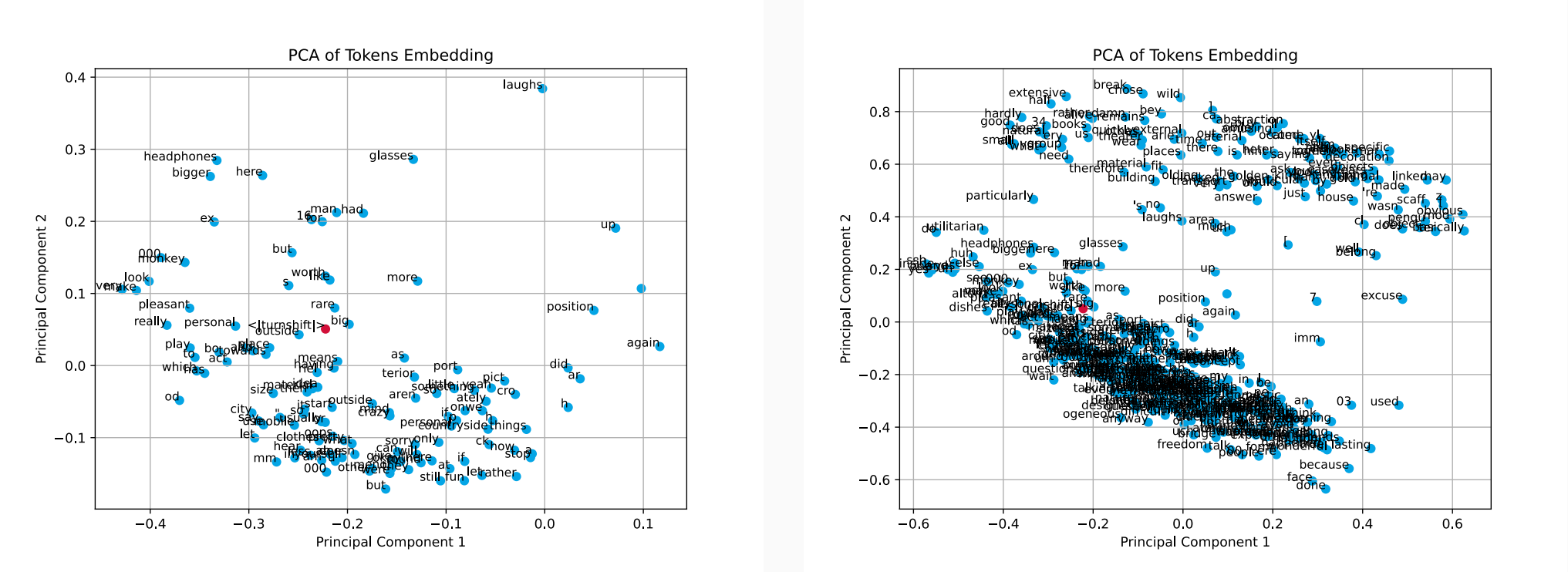


Fig. 4. – Prédiction de changement de tour sur une phrase (en pointillé le threshold d'acceptation de changement de tour du modèle)

Interprétation des prédiction : matrice d'embedding *(1 jour)*



(a) Graphe zoomé

(b) Graphe dézoomé

Fig. 5. – PCA des vecteurs d'embedding pour GPT2-large finetuné sur ChiCo anglophone (en rouge le vecteur <|turnshift|>)

Interprétation des prédiction : mécanisme d'attention *(2 jours)*

Layer Number	terday	we	met	in	the	park	<|turnshift|>
36	yes	<|turnshift|>	played	and	the	garden	and
32	ゼウス	evening	talked	again	the	garden	and
28	ゼウス	evening	talked	again	Paris	morning	where
24	ゼウス	evening	saw	again	front	morning	where
20	ゼウス	evening	've	with	front	same	where
16	ゼウス	evening	've	amorph	Prague	same	where
12	ゼウス	evening	're	amorph	Prague	same	lands
8	ゼウス	evening	ird	amorph	front	same	keepers
4	terday	evening	ird	ropolis	front	same	Meadows
0	yes	terday	we	met	in	the	park
Inputs	yes	terday	we	met	in	the	park

Tableau 3. – Utilisation de la *logit-lens* sur la phrase "yesterday we met in the park `<|turnshift|>`" (voir)

Comparaison Adulte/Enfant *(1 semaine)*

Pour les comparer : ajout de 2 tokens représentant enfant et adulte

⇒ `<|speaker1|>` et `<|speaker2|>`

```
<|speaker1|>
salut
<|speaker2|>
salut
<|speaker1|>
tu ne sais pas la dernière
<|speaker2|>
non
<|speaker1|>
aujourd'hui le facteur n'est pas passé on m'a même raconté qu'il ne passera jamais
non mais vraiment c'est honteux et moi qui attendais du courier
```

Comparaison Adulte/Enfant *(1 semaine)*

taille du contexte	75 tokens
threshold	5 %
modèle	gpt2-large (774M)
Adulte & Enfant	84.2 %
Adulte	86.2 %
Enfant	81.6 %

Tableau 4. – Taux de réussite de gpt2-large entraîné sur le corpus adulte/adulte en anglais selon l'âge de l'intervenant

Importance du gaze

ChiCa sans gaze	**ChiCa avec gaze**
84.1%	88.1 %

Tableau 5. – Taux de réussite de gpt2-large fintuné sur ChiCa anglais avec ou sans le gaze

10. Conclusion

10. Conclusion

Limites :

- Faible taille des datasets utilisés
- Manque de résultats d'interpétabilité $\Rightarrow$ diminue utilité théorique du modèle

Pistes d'améliorations :

- Production de datasets conversationels plus conséquents
- Plus de résultats d'iintérpretabilité entre adulte et enfant
- Utilisation de la tuned lens

Merci de votre attention !

11. Annexe

Répartition de mon temps

- **SEMAINE 1** : Lecture du papier de mon encadrant, Compréhension du modèle LSTM
- **SEMAINE 2** : Reproduction des résultats du papier de mon encadrant, Compréhension de l'architecture des transformers, Lecture de la littérature sur les tours de parole
- **SEMAINE 3** : Conception théorique du modèle de ChiCo, Mise en forme du dataset ChiCo, Implémentation du modèle, Premiers entraînements du modèle
- **SEMAINE 4** : Entraînement plus poussé du modèle, Entraînement d'un modèle sur ChiCo traduit en anglais, Premiers résultats d'interprétabilité
- **SEMAINE 5** : Test avec des LLM plus important en taille comme Claire-7B, Méthode pour évaluer enfants et adultes séparément, Entraînements et tests du modèle `<|speaker1|>/<|speaker2|>`
- **SEMAINE 6** : Conception d'une méthode afin d'incorporer le gaze dans l'architecture du transformer, Mise en forme du dataset ChiCa
- **SEMAINE 7** : Implémentation de GazeGPT, Tests et entraînement du modèle
- **SEMAINE 8** : Résultats d'interprétabilité sur GazeGPT, Travail de relecture sur le rapport avec mon encadrant

Explication mécanisme d'attention

Une tête fonctionne en utilisant les poids suivants :

- Une matrice de requêtes (*queries*) : W_Q
- Une matrice de clés (*keys*) : W_K
- Une matrice de valeurs (*values*) : W_V

A chaque token, on va associer une question $\vec{Q} = W_Q\overrightarrow{\text{token}}$ et une clé $\vec{K} = W_K\overrightarrow{\text{token}}$. Pour savoir si un token de clé $\vec{K}$ répond correctement à une question $\vec{Q}$ associée à un autre token, on calcule $\vec{K} \cdot \vec{Q}$: plus la valeur est grande, le mieux le token répond à la question.

Voici un exemple de ce calcul pour la suite de tokens suivante `[today, the sky, is]` :

Explication mécanisme d'attention

	$\overrightarrow{\text{today}}$ $\downarrow_{W_Q}$ $\vec{Q}_1$	$\overrightarrow{\text{the}}$ $\downarrow_{W_Q}$ $\vec{Q}_2$	$\overrightarrow{\text{sky}}$ $\downarrow_{W_Q}$ $\vec{Q}_3$	$\vec{\text{is}}$ $\downarrow_{W_Q}$ $\vec{Q}_4$
$\overrightarrow{\text{today}} \underset{W_K}{\rightarrow} \vec{K}_1$	$\vec{K}_1 \cdot \vec{Q}_1$	$\vec{K}_1 \cdot \vec{Q}_2$	$\vec{K}_1 \cdot \vec{Q}_3$	$\vec{K}_1 \cdot \vec{Q}_4$
$\overrightarrow{\text{the}} \underset{W_K}{\rightarrow} \vec{K}_2$	$\vec{K}_2 \cdot \vec{Q}_1$	$\vec{K}_2 \cdot \vec{Q}_2$	$\vec{K}_2 \cdot \vec{Q}_3$	$\vec{K}_2 \cdot \vec{Q}_4$
$\overrightarrow{\text{sky}} \underset{W_K}{\rightarrow} \vec{K}_3$	$\vec{K}_3 \cdot \vec{Q}_1$	$\vec{K}_3 \cdot \vec{Q}_2$	$\vec{K}_3 \cdot \vec{Q}_3$	$\vec{K}_3 \cdot \vec{Q}_4$
$\vec{\text{is}} \underset{W_K}{\rightarrow} \vec{K}_4$	$\vec{K}_4 \cdot \vec{Q}_1$	$\vec{K}_4 \cdot \vec{Q}_2$	$\vec{K}_4 \cdot \vec{Q}_3$	$\vec{K}_4 \cdot \vec{Q}_4$

Or, chose très importante pour l'étape d'entraînement du modèle, on ne souhaite pas qu'un token ait des informations sur des tokens dans le futur (`the` ne doit pas savoir qu'il y a un `is` deux tokens plus loin). On va donc *masquer* les résultats ne respectant pas ce critère (-∞ étant la pire réponse à une question) :

Explication mécanisme d'attention

	$\overrightarrow{\text{today}}$	$\overrightarrow{\text{the}}$	$\overrightarrow{\text{sky}}$	$\vec{\text{is}}$
	$\downarrow$ W_Q	$\downarrow$ W_Q	$\downarrow$ W_Q	$\downarrow$ W_Q
	$\vec{Q}_1$	$\vec{Q}_2$	$\vec{Q}_3$	$\vec{Q}_4$
$\overrightarrow{\text{today}} \underset{W_K}{\rightarrow} \vec{K}_1$	$\vec{K}_1 \cdot \vec{Q}_1$	$\vec{K}_1 \cdot \vec{Q}_2$	$\vec{K}_1 \cdot \vec{Q}_3$	$\vec{K}_1 \cdot \vec{Q}_4$
$\overrightarrow{\text{the}} \underset{W_K}{\rightarrow} \vec{K}_2$	-∞	$\vec{K}_2 \cdot \vec{Q}_2$	$\vec{K}_2 \cdot \vec{Q}_3$	$\vec{K}_2 \cdot \vec{Q}_4$
$\overrightarrow{\text{sky}} \underset{W_K}{\rightarrow} \vec{K}_3$	-∞	-∞	$\vec{K}_3 \cdot \vec{Q}_3$	$\vec{K}_3 \cdot \vec{Q}_4$
$\vec{\text{is}} \underset{W_K}{\rightarrow} \vec{K}_4$	-∞	-∞	-∞	$\vec{K}_4 \cdot \vec{Q}_4$

Après cela, on convertit ces nombres en une distribution de probabilité avec l'algorithme du soft-max (voir [1]) et on convertit chaque token en son vecteur de valeur associé :

Explication mécanisme d'attention

$$\left| \begin{array}{c} \overrightarrow{\text{today}} \underset{W_V}{\rightarrow} \vec{V}_1 \\ \overrightarrow{\text{the}} \underset{W_V}{\rightarrow} \vec{V}_2 \\ \overrightarrow{\text{sky}} \underset{W_V}{\rightarrow} \vec{V}_3 \\ \vec{\text{is}} \underset{W_V}{\rightarrow} \vec{V}_4 \end{array} \right|$$

Et ainsi, pour chaque token, on met à jour le vecteur associé :

$$\overrightarrow{\text{sky}} \mathrel{+}= 0 \cdot \vec{V}_1 + 0 \cdot \vec{V}_2 + \text{softmax}\left(\vec{K}_3 \cdot \vec{Q}_3\right) \cdot \vec{V}_3 + \text{softmax}\left(\vec{K}_3 \cdot \vec{Q}_4\right) \cdot \vec{V}_4$$

Expérience contrôle *(1 jour)*

Pour tester l'architecture :

Exemple bidon ou changement de tour et regard sont corrélés à 100%

Fin de phrase ⇒ regard sur l'interlocuteur

Résultat : **98.2%** de taux de réussite

L'architecture est capable d'extraire des informations utiles à la prédiction de tour dans la direction du regard

Bibliographie

[1] Wikipedia, « Softmax function — Wikipedia, The Free Encyclopedia ». 2024.

[2] A. Agrawal, J. Liu, K. Bodur, B. Favre, et A. Fourtassi, « Development of Multimodal Turn Coordination in Conversations: Evidence for Adult-like behavior in Middle Childhood ». PsyArXiv, mai 2023. doi: 10.31234/osf.io/h8j6x.

[3] K. Bodur, M. Nikolaus, F. Kassim, L. Prévot, et A. Fourtassi, « ChiCo: A Multimodal Corpus for the Study of Child Conversation », in *Companion Publication of the 2021 International Conference on Multimodal Interaction*, in ICMI '21 Companion. Montreal, QC, Canada: Association for Computing Machinery, 2021, p. 158-163. doi: 10.1145/3461615.3485399.

[4] E. Ekstedt et G. Skantze, « TurnGPT: a Transformer-based Language Model for Predicting Turn-taking in Spoken Dialog », in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, nov. 2020, p. 2981-2990. doi: 10.18653/v1/2020.findings-emnlp.268.

[5] A. Vaswani *et al.*, « Attention is All you Need », in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, et R. Garnett, Éd., Curran Associates, Inc., 2017, p. . [En ligne]. Disponible sur: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Bibliographie

[6] T. Wolf *et al.*, « Transformers: State-of-the-Art Natural Language Processing », in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, oct. 2020, p. 38-45. [En ligne]. Disponible sur: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[7] S. Grant, « Attention in transformers, visually explained | Chapter 6, Deep Learning ». [En ligne]. Disponible sur: https://www.youtube.com/watch?v=eMlx5fFNoYc

[8] L. Martin *et al.*, « CamemBERT: a Tasty French Language Model », in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[9] E. Almazrouei *et al.*, « Falcon-40B: an open large language model with state-of-the-art performance », 2023.

[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, et I. Sutskever, « Language Models are Unsupervised Multitask Learners », 2019.

[11] E. A. J. Sacks Harvey; Schegloff, « A Simplest Systematics for the Organization of Turn-Taking for Conversation », 1974.

[12] E. Baines et C. Howe, « Discourse topic management and discussion skills in middle childhood: The effects of age and task », *First Language*, vol. 30, n° 3–4, p. 508-534, 2010.

Bibliographie

[13] Z. Degutyte et A. Astell, « The role of eye gaze in regulating turn taking in conversations: a systematized review of methods and findings », *Frontiers in Psychology*, vol. 12, p. 616471, 2021.

[14] L. Bereska et E. Gavves, « Mechanistic Interpretability for AI Safety–A Review », *arXiv preprint arXiv:2404.14082*, 2024.

[15] V. Zouhar *et al.*, « A formal perspective on byte-pair encoding », *arXiv preprint arXiv:2306.16837*, 2023.

[16] N. Belrose *et al.*, « Eliciting latent predictions from transformers with the tuned lens », *arXiv preprint arXiv:2303.08112*, 2023.

[17] R. Sennrich, B. Haddow, et A. Birch, « Neural machine translation of rare words with subword units », *arXiv preprint arXiv:1508.07909*, 2015.

[18] Wikipedia, « Principal component analysis — Wikipedia, The Free Encyclopedia ». 2024.

[19] K. Meng, D. Bau, A. Andonian, et Y. Belinkov, « Locating and Editing Factual Associations in GPT ». [En ligne]. Disponible sur: https://arxiv.org/abs/2202.05262