Architecture Légère de Transformer pour la Reconnaissance d'Écriture Manuscrite

Killian Barrere

10 décembre 2021

Encadré par :
Yann Soullard, Aurélie Lemaitre, Bertrand Coüasnon
Équipe IntuiDoc, Univ. Rennes, CNRS, IRISA

État de l'art de la Reconnaisance d'Écriture : des RNN aux Transformers

Approche usuelle : Convolutional Recurrent Neural Networks (CRNN)

- CNN + RNN

$\Rightarrow$ **Mauvaise paraléllisation des RNN** qui limite la vitesse d'apprentissage

État de l'art de la Reconnaisance d'Écriture : des RNN aux Transformers

Approche usuelle : Convolutional Recurrent Neural Networks (CRNN)

- CNN + RNN

⇒ **Mauvaise paralélisation des RNN** qui limite la vitesse d'apprentissage

Approches Fully Convolutional Networks

- Composés majoritairement de CNN
- Pas de RNN

⇒ **Difficile d'apprendre des dépendances contextuelles larges**

État de l'art de la Reconnaisance d'Écriture : des RNN aux Transformers

Approche usuelle : Convolutional Recurrent Neural Networks (CRNN)

- CNN + RNN

⇒ **Mauvaise paralélisation des RNN** qui limite la vitesse d'apprentissage

Approches Fully Convolutional Networks

- Composés majoritairement de CNN
- Pas de RNN

⇒ **Difficile d'apprendre des dépendances contextuelles larges**

Couches Transformer / Attention à têtes multiples

- Capable d'apprendre des **dépendances contextuelles larges** et **paralélisable**

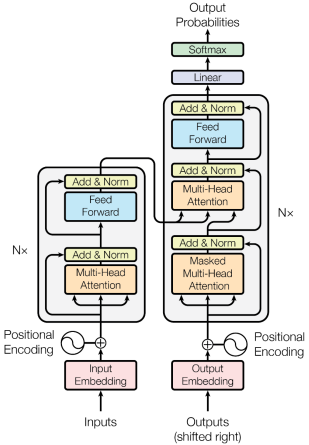
⇒ **Besoin de beaucoup de données** pour obtenir des résultats convenables

Thématiques abordées :

But : Reconnaissance de textes manuscrits anciens

Thématiques abordées :

But : Reconnaissance de textes manuscrits anciens ⇒ Transformers : Modélisation de la langue (LM)



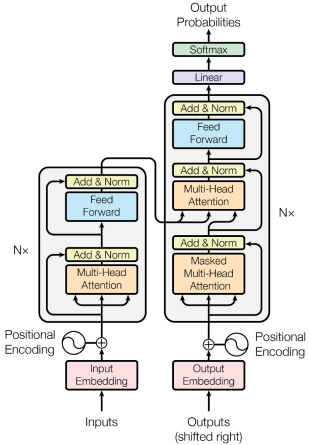
[Vaswani et al., 2017]

Thématiques abordées :

But : Reconnaissance de textes manuscrits anciens ⇒ Transformers : Modélisation de la langue (LM)

Idée : Modèle basé sur un Transformer :

- Reco. optique + Modélisation de la langue
- S'entraine de manière "**end-to-end**"



[Vaswani et al., 2017]

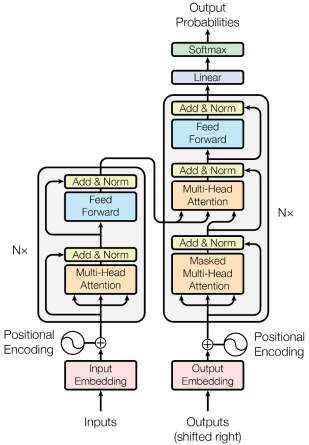
Thématiques abordées :

But : Reconnaissance de textes manuscrits anciens ⇒ Transformers : Modélisation de la langue (LM)

Idée : Modèle basé sur un Transformer :

- Reco. optique + Modélisation de la langue
- S'entraine de manière "**end-to-end**"

Problème : Besoin de beaucoup de données



[Vaswani et al., 2017]

Thématiques abordées :

But : Reconnaissance de textes manuscrits anciens ⇒ Transformers : Modélisation de la langue (LM)

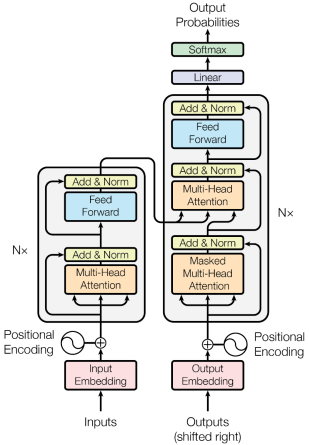
Idée : Modèle basé sur un Transformer :

- Reco. optique + Modélisation de la langue
- S'entraine de manière "**end-to-end**"

Problème : Besoin de beaucoup de données

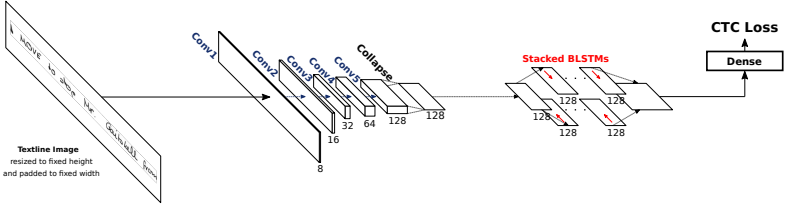
Architecture basée Transformer proposée :

- Architecture **légère**
- Utilisant une **loss hybride**

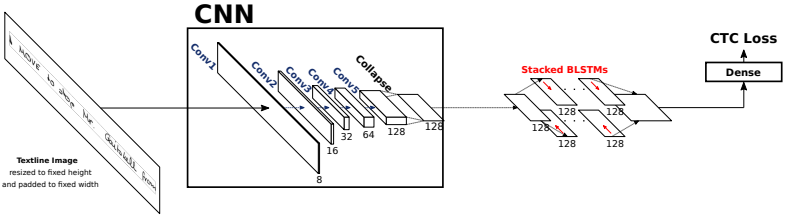


[Vaswani et al., 2017]

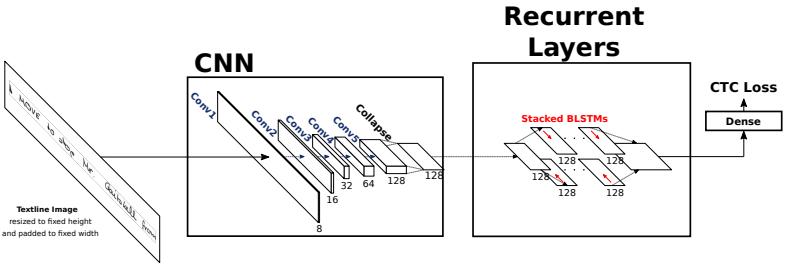
Du CRNN vers mon Architecture



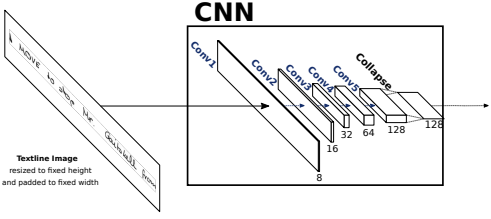
Du CRNN vers mon Architecture



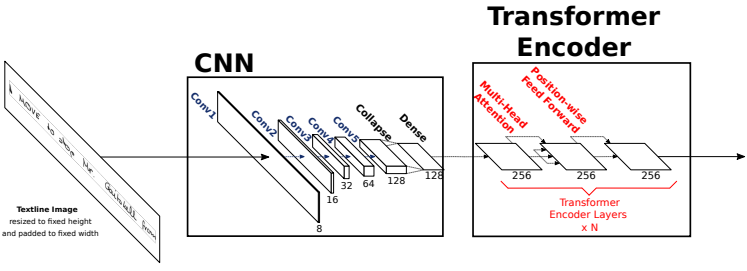
Du CRNN vers mon Architecture



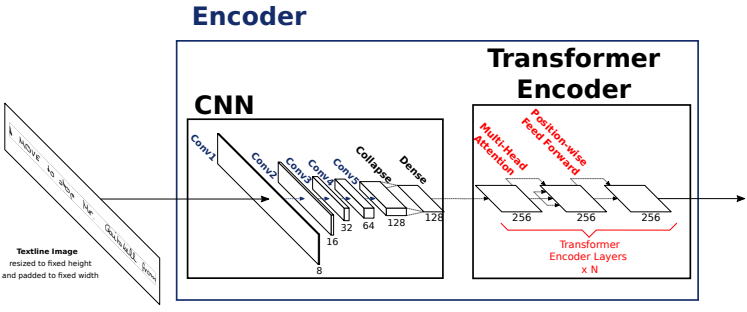
Du CRNN vers mon Architecture



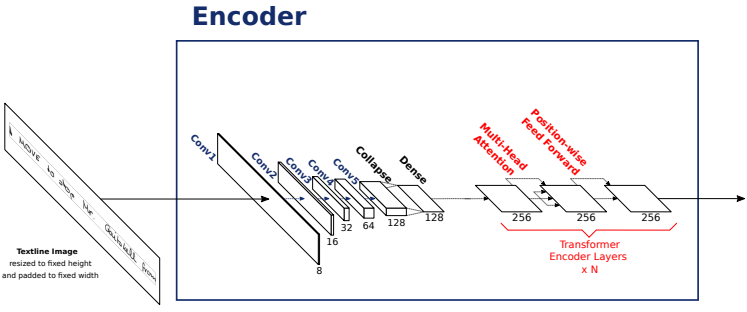
Du CRNN vers mon Architecture



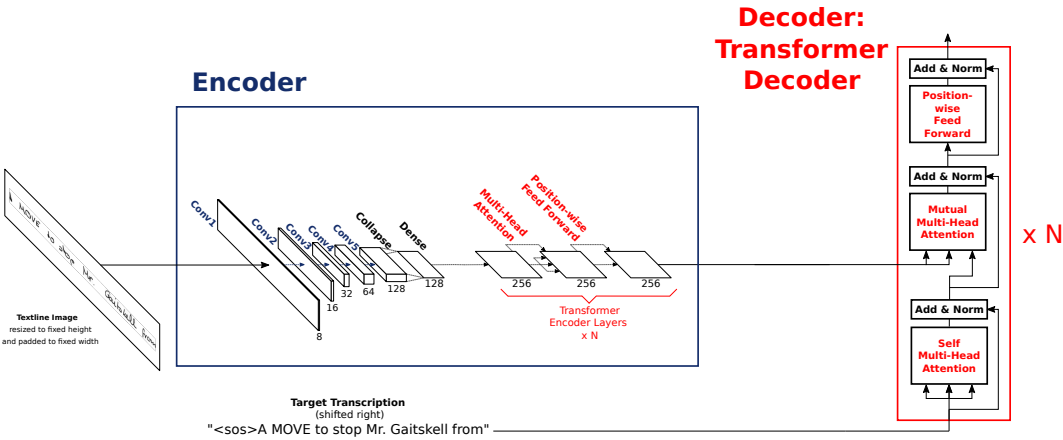
Du CRNN vers mon Architecture



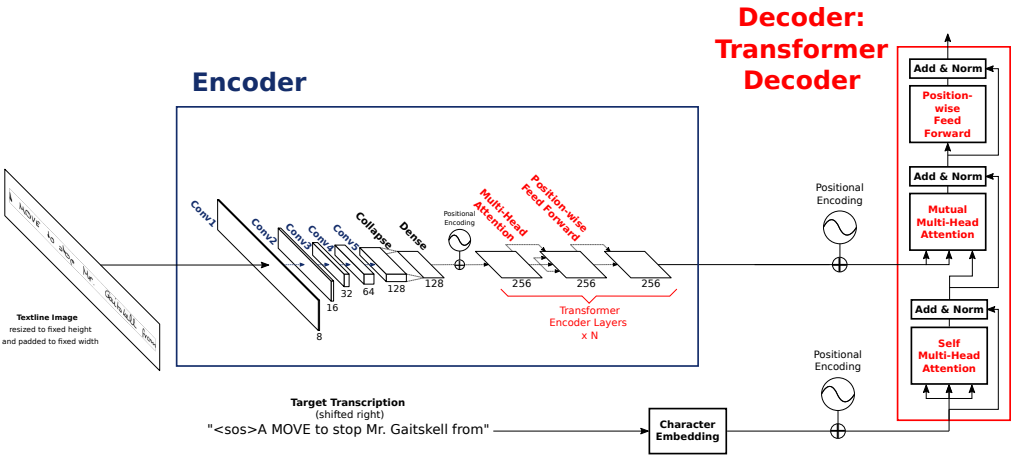
Du CRNN vers mon Architecture



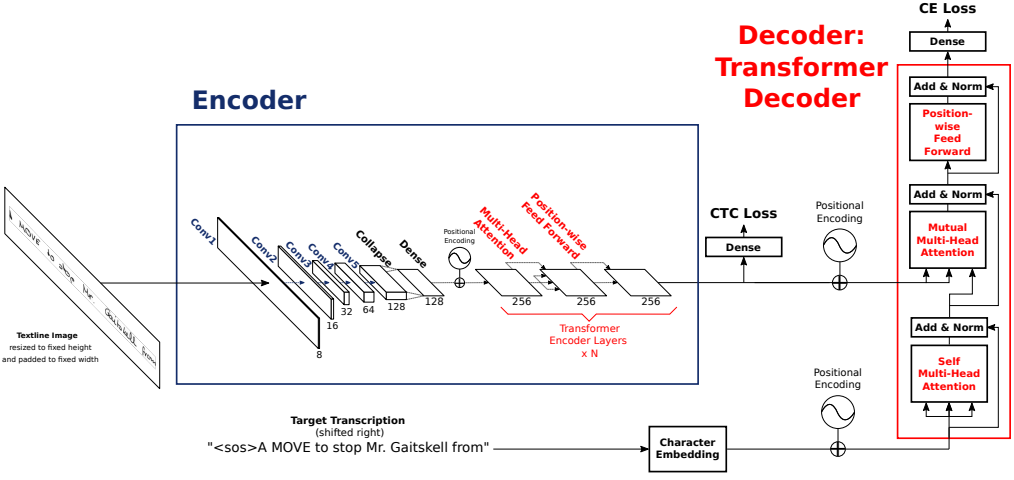
Du CRNN vers mon Architecture



Du CRNN vers mon Architecture



Du CRNN vers mon Architecture



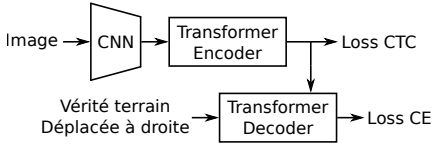
Comparaison de l'influence de la taille

But de l'expérience

On compare :

- La même architecture
- Avec **différentes tailles**
  - **Modèle léger : 7.7M paramètres**
  - Modèle lourd : 29.9M paramètres
- Sur IAM (Anglais Moderne)

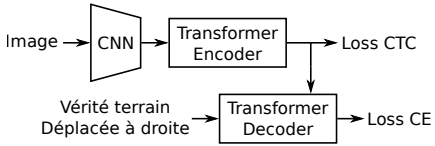
Loss Hybride



Loss Hybride [Michael et al., 2019] (Inspirée des réseaux Denses)

- Connectionist Temporal Classification (CTC) pour l'Encoder
- Cross Entropy (CE) pour le Decoder

Loss Hybride

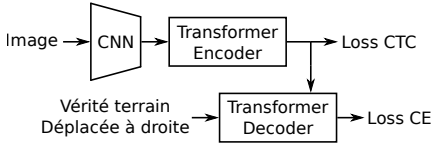


$$\mathcal{L} = \lambda \cdot \mathcal{L}_{CTC} + (1 - \lambda) \cdot \mathcal{L}_{CE}$$
$$\lambda = 0.5$$

Loss Hybride [Michael et al., 2019] (Inspirée des réseaux Denses)

- Connectionist Temporal Classification (CTC) pour l'Encoder
- Cross Entropy (CE) pour le Decoder

Loss Hybride



$$\mathcal{L} = \lambda \cdot \mathcal{L}_{CTC} + (1 - \lambda) \cdot \mathcal{L}_{CE}$$
$$\lambda = 0.5$$

Loss Hybride [Michael et al., 2019] (Inspirée des réseaux Denses)

- Connectionist Temporal Classification (CTC) pour l'Encoder
- Cross Entropy (CE) pour le Decoder

Intérêt

- Aider l'entraînement des couches plus profondes
- Convergence beaucoup plus rapide

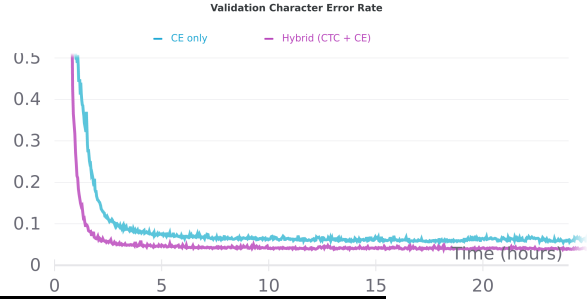
$\Rightarrow$ **Amélioration des résultats**

Comparaison des fonctions de coûts

But de l'expérience

On étudie :

- La même architecture (mon Transformer léger)
- **L'intérêt de la loss hybride**
- Sur IAM (Anglais Moderne)



Model	Validation		Test	
	CER	WER	CER	WER
CE seule	7.89	22.36	10.29	26.36
Hybride (CTC + CE)	4.11	15.27	5.70	18.86

Comparaison à l'État de l'art sur IAM

Model Encoder	# params.	Additional information	Test CER	Test WER
CRNN + LSTM [Michael et al., 2019]			5.24	
FCN [Yousef et al., 2020]	3.4M		**4.9**	
VAN (line level) [Coquenet et al., 2020]	1.7M		4.95	16.24
Transformer [Kang et al., 2020]	100M	-	7.62	24.54
		synth. data	4.67	**15.45**
FPHR Transformer [Singh et al., 2021]	28M	synth. data	6.5	
Bidi. Transformer [Wick et al., 2021]			5.67	
Our Transformer-based	7.7M		5.70	18.86

Comparaison à l'État de l'art sur IAM

Model Encoder	# params.	Additional information	Test CER	Test WER
CRNN + LSTM [Michael et al., 2019]			5.24	
FCN [Yousef et al., 2020]	3.4M		**4.9**	
VAN (line level) [Coquenet et al., 2020]	1.7M		4.95	16.24
Transformer [Kang et al., 2020]	100M	-	7.62	24.54
		synth. data	4.67	**15.45**
FPHR Transformer [Singh et al., 2021]	28M	synth. data	6.5	
Bidi. Transformer [Wick et al., 2021]			5.67	
Our Transformer-based	7.7M		5.70	18.86

Comparaison à l'État de l'art sur IAM

Model Encoder	# params.	Additional information	Test CER	Test WER
CRNN + LSTM [Michael et al., 2019]			5.24	
FCN [Yousef et al., 2020]	3.4M		**4.9**	
VAN (line level) [Coquenet et al., 2020]	1.7M		4.95	16.24
Transformer [Kang et al., 2020]	100M	-	7.62	24.54
		synth. data	4.67	**15.45**
FPHR Transformer [Singh et al., 2021]	28M	synth. data	6.5	
Bidi. Transformer [Wick et al., 2021]			5.67	
Our Transformer-based	7.7M		5.70	18.86

Comparaison à l'État de l'art sur IAM

Model Encoder	# params.	Additional information	Test CER	Test WER
CRNN + LSTM [Michael et al., 2019]			5.24	
FCN [Yousef et al., 2020]	3.4M		**4.9**	
VAN (line level) [Coquenet et al., 2020]	1.7M		4.95	16.24
Transformer [Kang et al., 2020]	100M	-	7.62	24.54
		synth. data	4.67	**15.45**
FPHR Transformer [Singh et al., 2021]	28M	synth. data	6.5	
Bidi. Transformer [Wick et al., 2021]			5.67	
Our Transformer-based	7.7M		5.70	18.86

Conclusion

- **Architecture de Transformer légère**, peu coûteuse
  $\Rightarrow$ Plus rapide à entraîner que les autres Transformers
- Entrainée avec une **loss Hybride** pour améliorer les performances
- **Des résultats corrects** sans données ajoutées
  $\Rightarrow$ De la marge de progression

Conclusion

- **Architecture de Transformer légère**, peu coûteuse
  ⇒ Plus rapide à entraîner que les autres Transformers
- Entrainée avec une **loss Hybride** pour améliorer les performances
- **Des résultats corrects** sans données ajoutées
  ⇒ De la marge de progression

- Le modèle semble sur-apprendre

⇒ Besoin de **plus de données** pour obtenir de meilleurs résultats
⇒ Besoin de plus de techniques de **régularisation**

Travaux futurs

Documents anciens

- Capacité de **modélisation de langue** des transformers cruciale pour de l'ancien
- **Très peu de données annotées** $\Rightarrow$ Intérêt d'utiliser une petite architecture

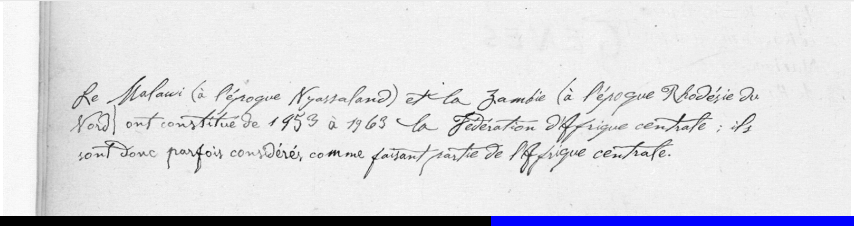
Travaux futurs

Documents anciens

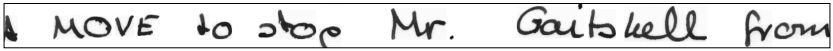
- Capacité de **modélisation de langue** des transformers cruciale pour de l'ancien
- **Très peu de données annotées** ⇒ Intérêt d'utiliser une petite architecture

Besoin de plus de données ?

- **Génération de données synthétiques**
  ⇒ Premiers résultats encourageants



A l'entrainement : Decoding en paralèlle et Teacher Forcing



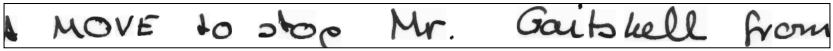
<sos>A MOVE to stop Mr. Gaitskell from

En paralèlle grâce à du Masking

<sos> → A
<sos>A → A_
<sos>A_ → A_N
<sos>A_M → A_MO
<sos>A_MO → A_MOV
⋮ ⋮ ⋮
<sos>A_MOVE_to_stop_Mr._Gaitskell_from → A_MOVE_to_stop_Mr_Gaitskell_from<eos>

<sos>A_MOVE_to_stop_Mr._Gaitskell_from → A_NOVE_to_stop_Mr_Gaitskell_from<eos>

A l'inférence : Decoding séquentiel



Decoding
Séquentiel

<sos> → A
<sos>A → A_
<sos>A_ → A_N
<sos>A_N → A_NO
<sos>A_NO → A_NOV
⋮ ⋮ ⋮
<sos>A_NOVE_to_stop_Mr._Gaitswell_from → A_NOVE_to_stop_Mr_Gaitswell_from<eos>

→ A_NOVE_to_stop_Mr_Gaitswell_from<eos>

CER and WER

Definition

- CER: Character Error Rate
- WER: Word Error Rate
- Edit distance between prediction and groundtruth

$$CER = \frac{I+D+S}{N}$$

I Number of Insertions
D Number of Deletions
S Number of Substitutions
N Length of the groudtruth