

Introduction

Nous souhaitons livrer un produit utilisable en temps réel par l'entremise de ce projet. Les projets académiques habituellement réalisés se concentrent sur les l'optimisation et la comparaison de modèles, mais nous n'avons jamais eu la chance de terminer la chaîne d'un projet et d'utiliser les prédictions des modèles développés en **temps réel**.

Motivations :

- Application des techniques de **traitement d'images** à partir d'un grand corpus de données.
- Déploiement complet d'un modèle prédictif pour usage par des non-experts.

Buts :

- Apprendre à traiter efficacement de grandes quantités de données.
- Se familiariser avec les outils d'**infonuagique** des grands joueurs pour projets à grande échelle.
- Développer une **interface graphique** permettant à des usagers de tester notre modèle en temps réel.
- Couvrir l'ensemble de la chaîne d'un projet en intelligence artificielle du traitement des données à l'utilisation du modèle dans un contexte de vie réelle.

Contraintes

Plusieurs contraintes dans la réalisation de ce projet nous ont empêchées de tester toutes les avenues que nous avons initialement envisagées.

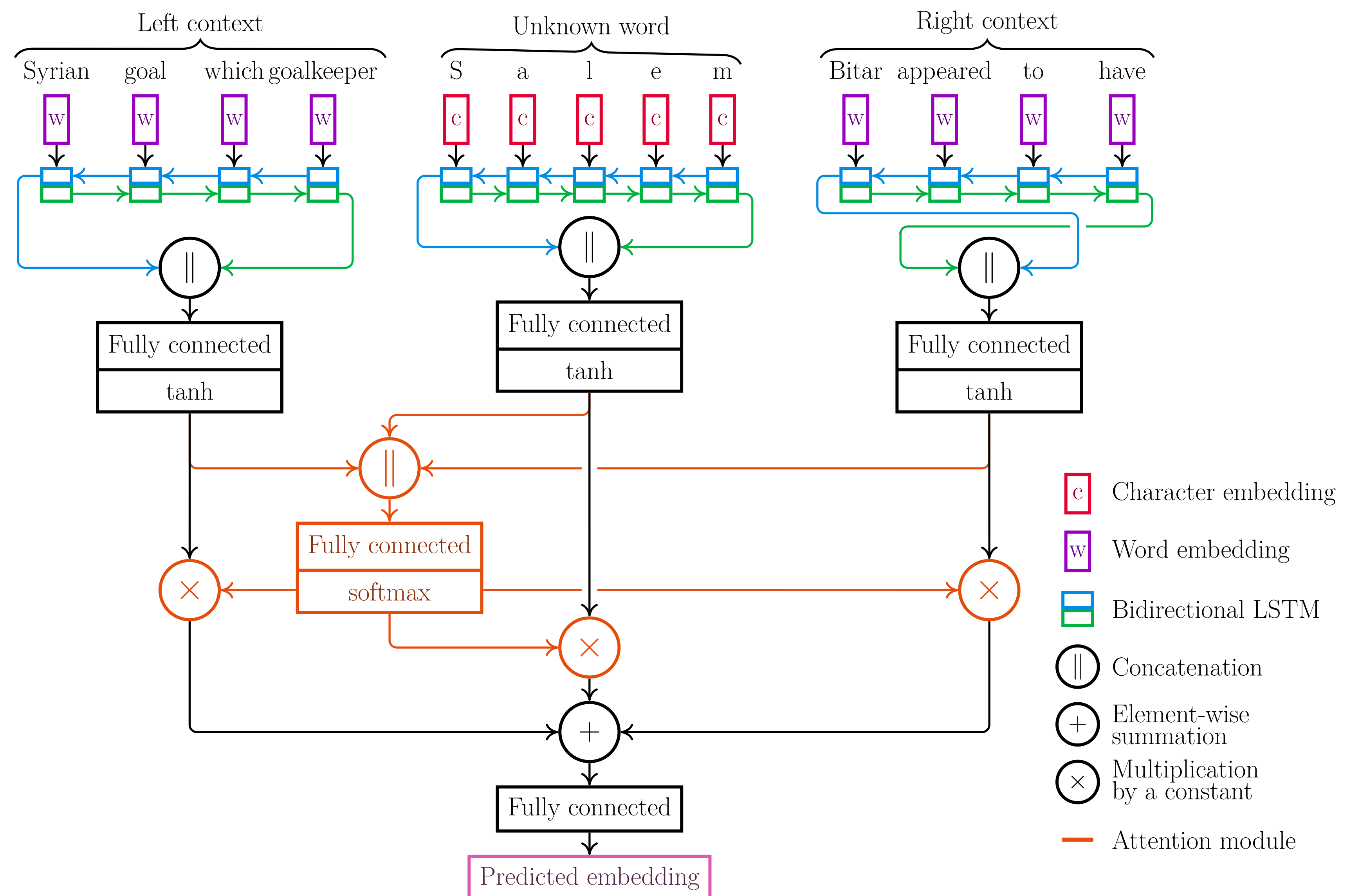
Contraintes computationnelles :

- Enjeux de mémoire :
 - Incapacité à charger l'ensemble du jeu de données en **mémoire**.
 - Conservation des images en **format vectoriel** pour limiter espace de stockage nécessaire.
- Enjeux de performance :
 - Temps de traitement très important nous limite dans le nombre de modèles et de jeux d'hyperparamètres qu'on peut tester.
 - Impossibilité d'entraîner les modèles sur nos postes personnels. Nous devons utiliser des solutions d'**infonuagique** qui peuvent être coûteuses et complexe à utiliser avec un jeu de données aussi important.

Contraintes de l'interface graphique :

- Perte de l'**information temporelle** des images vectorielles.
- Impossibilité d'utiliser **différents canaux** pour simuler l'évolution temporelle du dessin.
- Perte de la séparation du dessin par trait.

Structure du réseau



The net consists in 3 bi-LSTM taking as input the left context, the right context and the word characters. An attention module ponderates their outputs which are then combined in a last fully connected layer.

Transformations

Entity	Ponderation			Examples
	Word	Left	Right	
PER	0.19	0.49	0.32	in sentencing darrel <u>voeks</u> , 38 , to a 10-year prison term. <BOS> australian parliamentarian john langmore has formally announced that he will not be re-elected. <BOS> rtrs - australian mp john langmore formally announced that he will not be re-elected.
PER	0.15	0.59	0.26	
PER	0.15	0.61	0.24	
PER	0.15	0.69	0.16	
ORG	0.22	0.46	0.32	the number of plastic surgeries in [...] the brazilian plastic surgeon dr. roberto fernandes to increase them in the united states , ” sbcp vice-president roberto fernandes some residents of the kazanluk area are moslems who are muslims at a mosque in the central bulgarian town of kazanluk , ca freestyle skiing-world cup aerials res
ORG	0.28	0.23	0.49	
LOC	0.16	0.22	0.62	
LOC	0.20	0.47	0.33	
MISC	0.68	0.11	0.21	the franco-african summit decided to send a mission to the
MISC	0.42	0.18	0.40	

Qualitative example on several OOV words (underlined). We can see that depending on the context and the target, the weights may shift drastically.

Description des données

Le jeu de données de Quick, Draw! est un sous-échantillon du jeu de données de Google AI contenant plus d'un milliard d'images. Voici quelques caractéristiques du jeu de données utilisé pour l'entraînement du modèle :

- Jeux de données bruts et simplifiés disponibles (le jeu de données simplifié élimine des points inutiles qui se trouvent sur une même droite).
- **345 classes**.
- **≈ 150 000 images/classe**.
- **24,4 Gb** de données compressées en format .csv.

Utilisation des données compressées pour le projet pour limiter les enjeux de mémoire autant que possible sous hypothèse que les performances du modèle n'en souffriraient pas.

Performances

Task	Tag	Ex.	Ponderation		
			Word	Left	Right
NER	O	1039	0.81	0.08	0.11
	B-PERS	63	0.21	0.31	0.49
	I-PER	119	0.16	0.52	0.32
	B-ORG	40	0.26	0.30	0.44
	I-ORG	3	0.27	0.31	0.42
	B-LOC	13	0.23	0.30	0.47
	I-LOC	2	0.16	0.48	0.36
	B-MISC	47	0.40	0.21	0.39
POS	I-MISC	5	0.41	0.26	0.33
	NNP	308	0.29	0.31	0.40
	NN	46	0.45	0.20	0.35
	CD	827	0.86	0.05	0.09
	NNS	23	0.37	0.24	0.39
	JJ	100	0.49	0.15	0.36

Average weights assigned to word's characters, left context and right context by the attention mechanism. We can clearly see the shift of attention according to the target entity. We also observe that the attention depends on the task at hand.

Conclusion

Discussion :

- **Morphology** and **context** help predict useful embeddings.
- **The attention mechanism works** : depending on the task, the network will use either more the context or the morphology to generate an embedding.

Future works :

- Apply the **attention mechanism on each character of the OOV word and each word of the context** instead of using the hidden state of the respective elements only.
- Test our attention model in **different languages** and on other NLP tasks, such as **machine translation**.