

## Introduction

Nous souhaitons livrer un produit utilisable en temps r  el par l'entremise de ce projet. Les projets acad  miques habituellement r  alis  s se concentrent sur les l'optimisation et la comparaison de mod  les, mais nous n'avons jamais eu la chance de terminer la cha  ne d'un projet et d'utiliser les pr  dictions des mod  les d  velopp  s en **temps r  el**.

### Motivations :

- Application des techniques de **traitement d'images**    partir d'un grand corpus de donn  es.
- D  ploiement complet d'un mod  le pr  dictif pour usage par des non-experts.

### Buts :

- Apprendre    traiter efficacement de grandes quantit  s de donn  es.
- Se familiariser avec les outils d'**infonuagique** des grands joueurs pour projets    grande   chelle.
- D  velopper une **interface graphique** permettant    des usagers de tester notre mod  le en temps r  el.
- Couvrir l'ensemble de la cha  ne d'un projet en intelligence artificielle du traitement des donn  es    l'utilisation du mod  le dans un contexte de vie r  elle.

## Contraintes

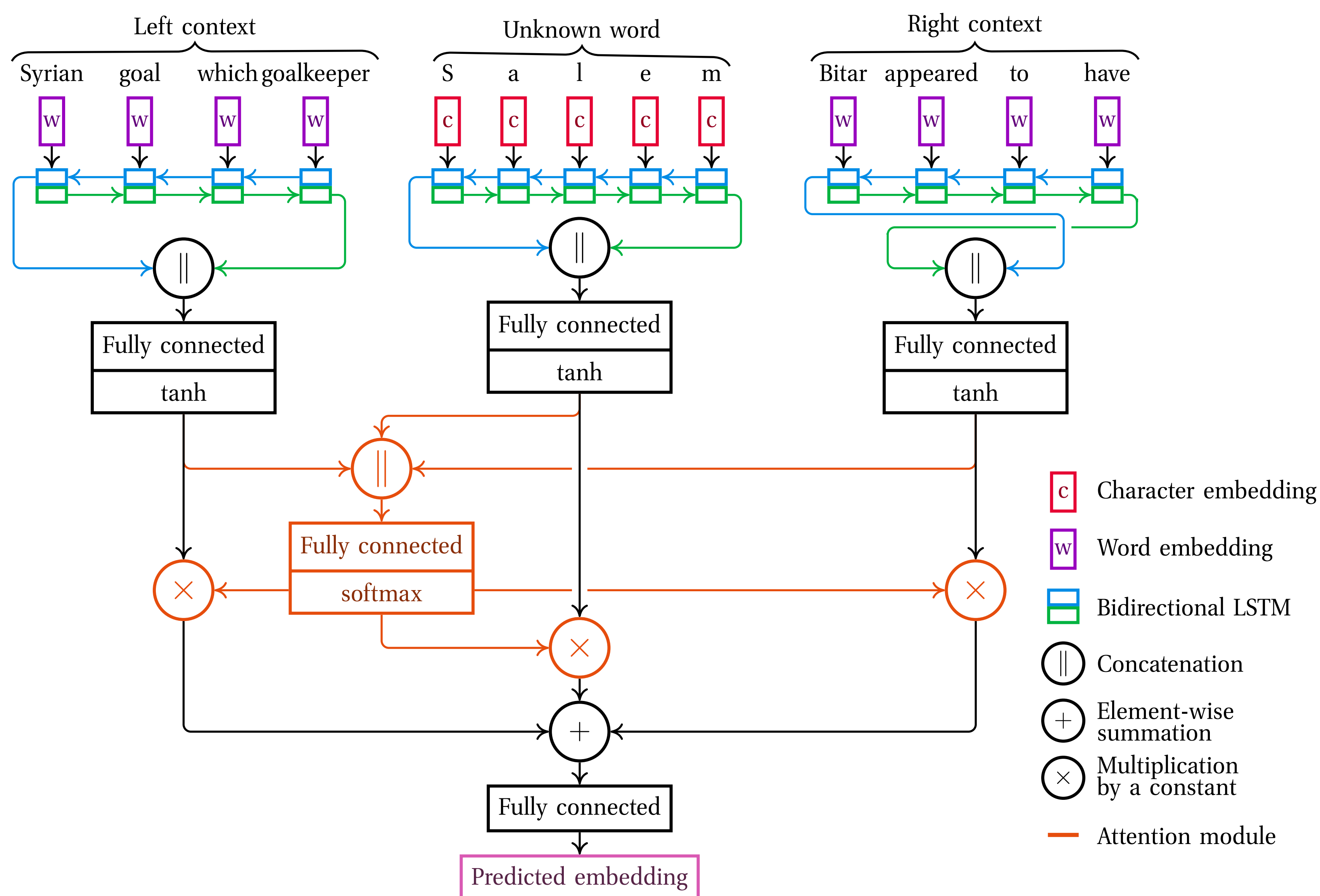
Plusieurs contraintes dans la r  alisation de ce projet nous ont emp     es de tester toutes les avenues que nous avons initialement envisag  es. **Contraintes computationnelles :**

- Enjeux de m  moire :
  - Incapacit      charger l'ensemble du jeu de donn  es en **m  moire**.
  - Conservation des images en **format vectoriel** pour limiter espace de stockage n  cessaire.
- Enjeux de performance :
  - Temps de traitement tr  s important nous limite dans le nombre de mod  les et de jeux d'hyperparam  tres qu'on peut tester.
  - Impossibilit   d'entra  ner les mod  les sur nos postes personnels. Nous devons utiliser des solutions d'**infonuagique** qui peuvent   tre co  teuses et complexe    utiliser avec un jeu de donn  es aussi important.

### Contraintes de l'interface graphique :

- Perte de l'**information temporelle** des images vectorielles.
- Impossibilit   d'utiliser **diff  rents canaux** pour simuler l'  volution temporelle du dessin.
- Perte de la s  paration du dessin par trait.

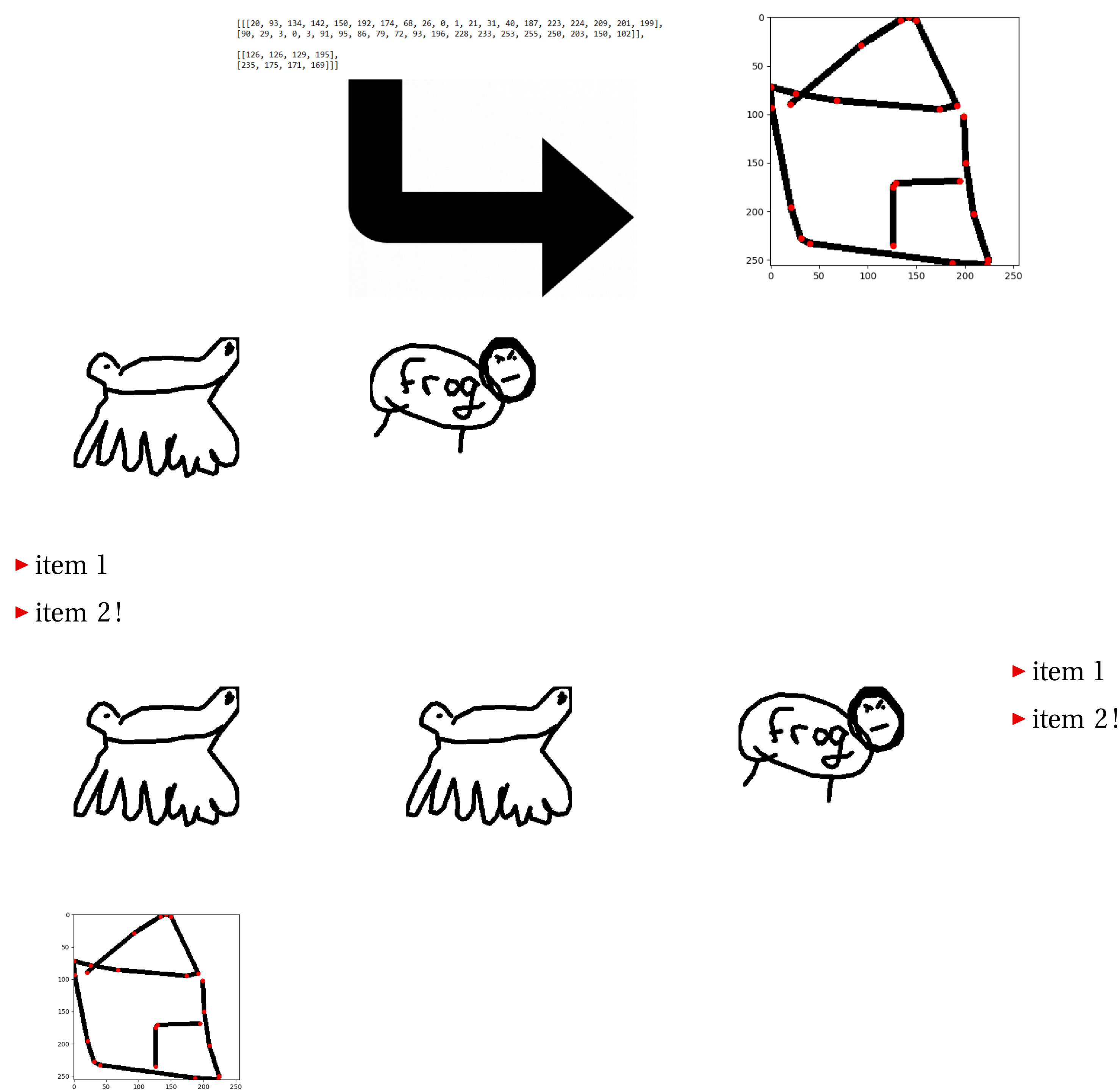
## Structure du r  seau



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

### Transformation de vecteurs de position en photos



Yesterday,  
all my troubles seemed so far away  
Now it looks as though they're here to stay  
Oh, I believe in yesterday.  
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 donn  es

Le jeu de donn  es de Quick, Draw! est un sous-  chantillon du jeu de donn  es de Google AI contenant plus d'un milliard d'images. Voici quelques caract  ristiques du jeu de donn  es utilis   pour l'entra  nement du mod  le :

## Performances

Task	Tag	Ex.	Ponderation		
			Word	Left	Right
	O	1039	<b>0.81</b>	0.08	0.11
	B-PERS	63	0.21	0.31	<b>0.49</b>

## 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.