

# Normaliser la langue

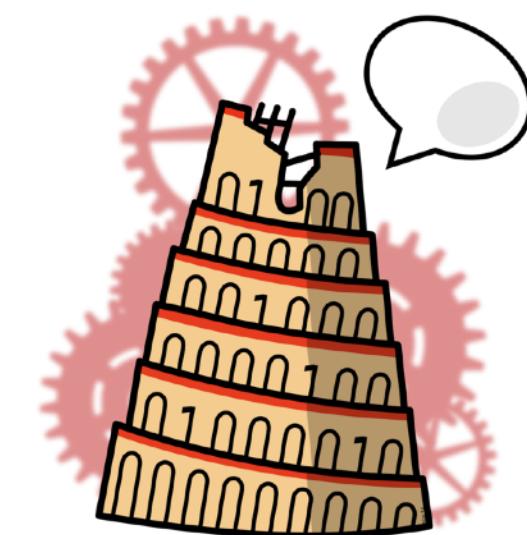
Rachel Bawden

Inria Paris (équipe-projet ALMAnaCH)

*Philologie computationnelle : au delà de l'encodage du texte*

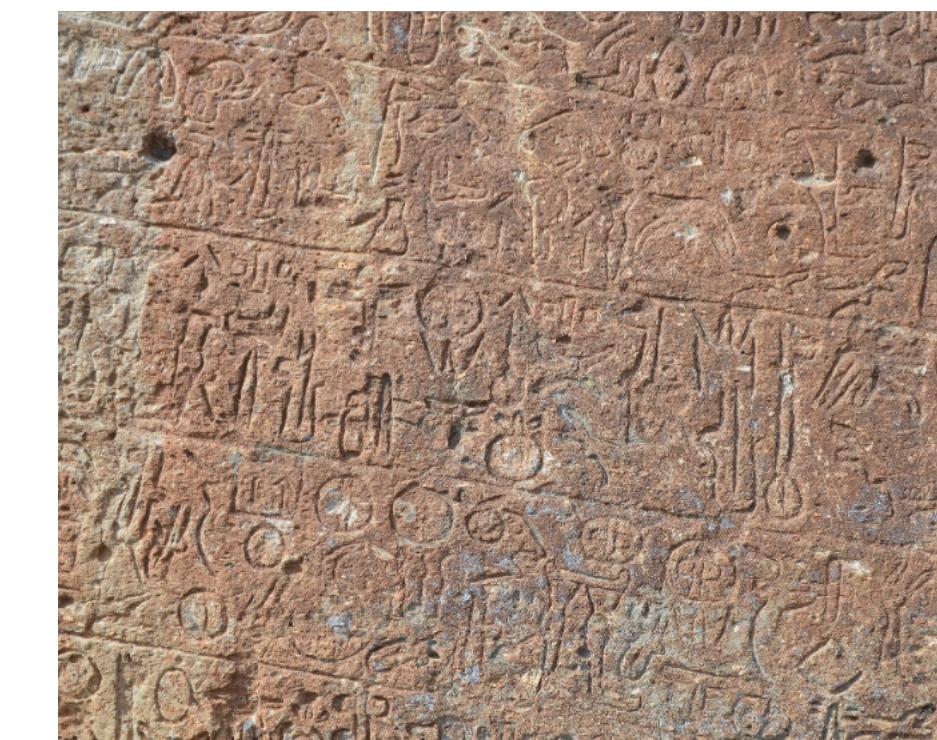
3/12/2021

Inria



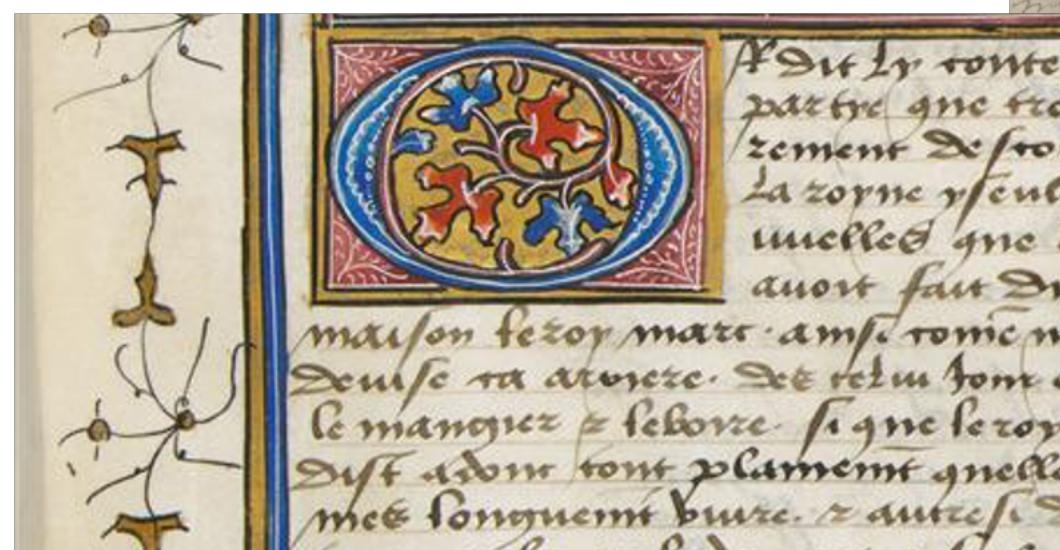
# Normalisation - quoi et pourquoi ?

## Le défi de la variation



Le moulin 21 Decembre 1915

Mon très cher bien aimé Pierre  
Nous voila après dîner jeté  
dans son petit sommeil Trop et il me manque  
toujours de tout les jours signifier aux cochons  
les nuits et Mois et l'oisive se prépare pour  
aller à planter un ou deux de nos vignes Il fait  
tous magnifique le soleil est très riche et le tra  
verso est fait de tue sur une mardi et au  
beau temps elle sont dédiée tout d'un coup sa  
peut tout le monde est occupé et moi je veux pas  
un petit instant au sein de ma douce compagnie  
moment doux à mon cœur Je suis fort dans mes  
mains j'ouvrirai dit que bien nous a  
tous les deux de bonnes bi



A dit le conte en rete  
partre que trop fut du  
rement de confortee.  
La zorne rfeult des no  
uvelles que auz  
auoit fait dñ  
maison lezoy març amf. romie no  
deusse na armere. des retz tom e  
le manquer lezoy signe lezoy  
dist ador tout plament quelle  
mee longuent vire. et autres

I don't understand why there are Toffee Pennies in Quality Street. Are they for people who don't like chocolate but really fancy buying a tin of Quality Street?

7:35 pm · 9 Nov 2021 · Twitter for iPhone

42 Retweets 35 Quote Tweets 2,541 Likes

...

Richard Osman ✅ @richardosman · 2h

Replying to @richardosman

Am shocked and saddened by some of the pro-Toffee Penny responses.

...

...

...

...

...

...

...

...

...



Article Talk

Not log

Read Edit

## Machine translation

From Wikipedia, the free encyclopedia

lation on Wikipedia, see [Wikipedia:Content translation tool](#).

sometimes referred to by the abbreviation **MT**<sup>[1]</sup> (not to puter-aided translation, machine-aided human translation on), is a sub-field of computational linguistics that f software to translate text or speech from one language

performs mechanical substitution of words in one another, but that alone rarely produces a good translation if whole phrases and their closest counterparts in the eded. Not all words in one language have equivalent uage, and many words have more than one meaning.

with **corpus** statistical and **neural** techniques is a rapidly- ading to better translations, handling differences in nslation of **idioms**, and the isolation of cation]

# Normalisation - quoi et pourquoi ?

- **Un défi important dans le développement d'outils de TAL : la variation dans les textes (synchronique/diachronique)**
  - Variation orthographique, linguistique, graphique
- **Normaliser = transformer un texte pour qu'il adhère à certaines normes**
  - Rend l'analyse systématique des textes plus facile
  - Rend plus facile l'utilisation de ressources entraînées sur les textes qui adhèrent à ces normes

# Normalisation - quoi et pourquoi ?

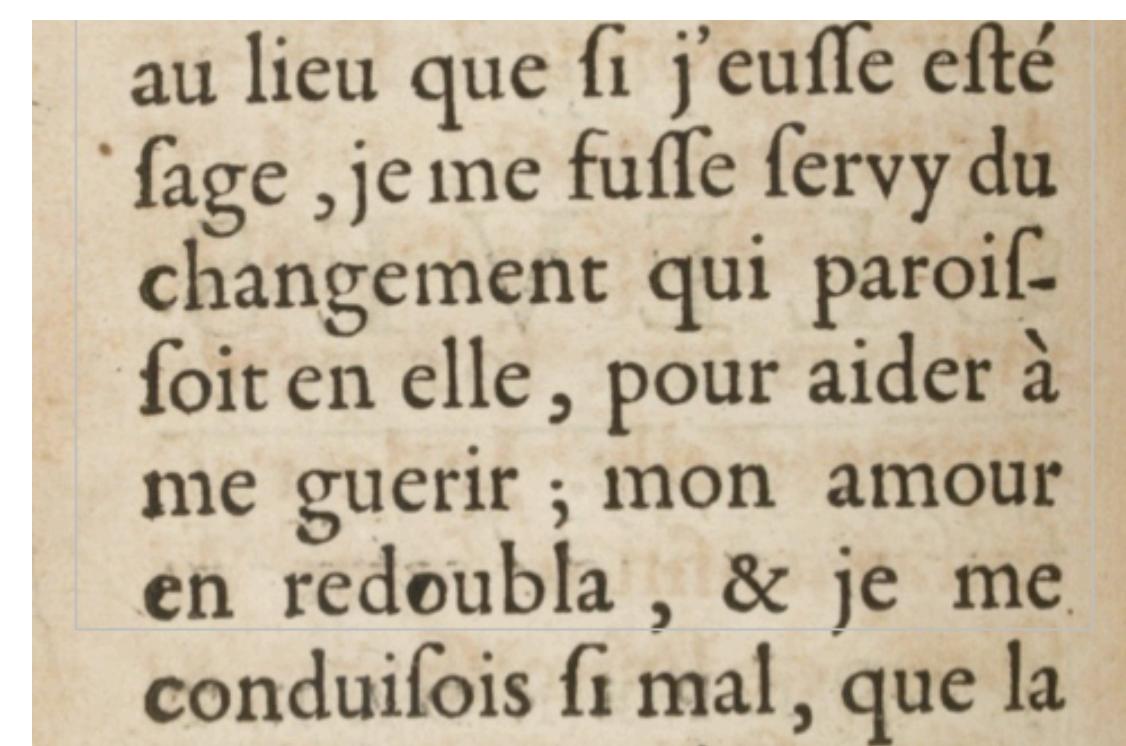
- Dans d'autres travaux : traduction automatique de textes des réseaux sociaux

:")) ofcfc n i cAN TELL HOW MUCH TIME N EFFORT u put into them cuz  
they always turn out lookin frikKIN ??4!4 AMAZING even when u think  
they arent ehHEHEHE LIKE EVEN IF ISS TERRAFORMING N NOT LIKE ?? A  
BUILDING OR A CASTLE OR A MONUMENT OR A VILLAGE OR WOTEVA,  
ITS ALWAYS !2!!!2!

hELLO OOMF 😷 😷 IM VERY LOUD AND USES EMOJIS ALOT ALOT AND I  
CURSE !! !! R U OK W THAT 😬 😬

mais tell them genre tu bosses pour toi pas pour les autres wtf encore  
2/3fois je dis pas mais là elles font 0 effort en cours ou ça se passe cmt?

- Ce tutoriel : normalisation du français moderne (du XVII<sup>ème</sup> siècle)



# Normalisation - comment ?

- Approches à base de règles
- Approches orientées données (*data-driven*)
  - À partir de données parallèles (textes d'origine et leurs versions normalisées par des experts), apprendre automatiquement les correspondances en s'appuyant sur les statistiques
- Plusieurs méthodes existent. Dans ce tutoriel je vais parler d'une approche utilisant la traduction automatique neuronale

# Plan

## **1. La normalisation du français moderne comme une tâche de traduction**

## **2. Partie technique**

1. Rappel sur les *word embeddings*
2. Explication des architectures de traduction

## **3. Partie expérimentale**

1. Expériences sur la normalisation du français moderne
2. Découvrir les changements linguistiques/graphiques au cours du XVII<sup>ème</sup> siècle

# 1. La normalisation du français moderne comme une tâche de traduction

# La normalisation du français moderne comme une tâche de traduction

- Ibid, 32, Il falloit se réjoûir parce que vostre frere estoit mort, & il eft reffuscité;
- SCavantes Sœurs, arbitres de la Scene,
- Obtenir encore ce poinct,
- Il eft vray qu'il ne fut pas long temps fans fçauoir qu'Horace n'estoit point mort:...
- Vouloient moins de savoir & plus d'amusemens,
- Ibid, 32, Il fallait se réjouir parce que votre frère était mort, et il est ressuscité;
- Savantes Sœurs, arbitres de la Scène,
- Obtenir encore ce point,
- Il est vrai qu'il ne fut pas longtemps sans savoir qu'Horace n'était point mort:
- Voulaient moins de savoir et plus d'amusement

# La normalisation du français moderne comme une tâche de traduction

- Ibid, 32, Il falloit se réjouir parce que vostre frere estoit mort, & il est ressuscité;
- SCavantes Sœurs, arbitres de la Scène,
- Obtenir encore ce point,
- Il est vray qu'il ne fut pas long temps fans fçauoir qu'Horace n'estoit point mort:...
- Voulloient moins de savoir & plus d'amusemens,
- Ibid, 32, Il fallait se réjouir parce que votre frère était mort, et il est ressuscité;
- Savantes Sœurs, arbitres de la Scène,
- Obtenir encore ce point,
- Il est vrai qu'il ne fut pas longtemps sans savoir qu'Horace n'était point mort:
- Voulaienr moins de savoir et plus d'amusement

# La normalisation du français moderne comme une tâche de traduction

Pourquoi normaliser vers le français contemporain ?

- Nous avons besoin de choisir un norme
- Il est plus facile de normaliser vers le français contemporain pour des annotateurs (et il y a une norme reconnue)
- Ça nous permet d'appliquer ensuite des modèles et des outils adaptés pour le français contemporain (pour lequel il y a plus de données)

# La normalisation du français moderne comme une tâche de traduction

*Il eſt vray qu'il ne fut pas **long temps** fans fçauoir qu'Horace n'**eſtoit** point mort:*



*Il est vrai qu'il ne fut pas **longtemps** sans savoir qu'Horace n'**était** point mort:*

- Transformation d'une séquence de texte en une autre séquence de texte
- Plus facile que la traduction automatique, car les deux « langues » sont très proches et il n'y a pas de changement dans l'ordre des mots, mais permet de changer le nombre de mots (*long temps* > *longtemps*)
- Permet d'exploiter la puissance des modèles de traduction état de l'art

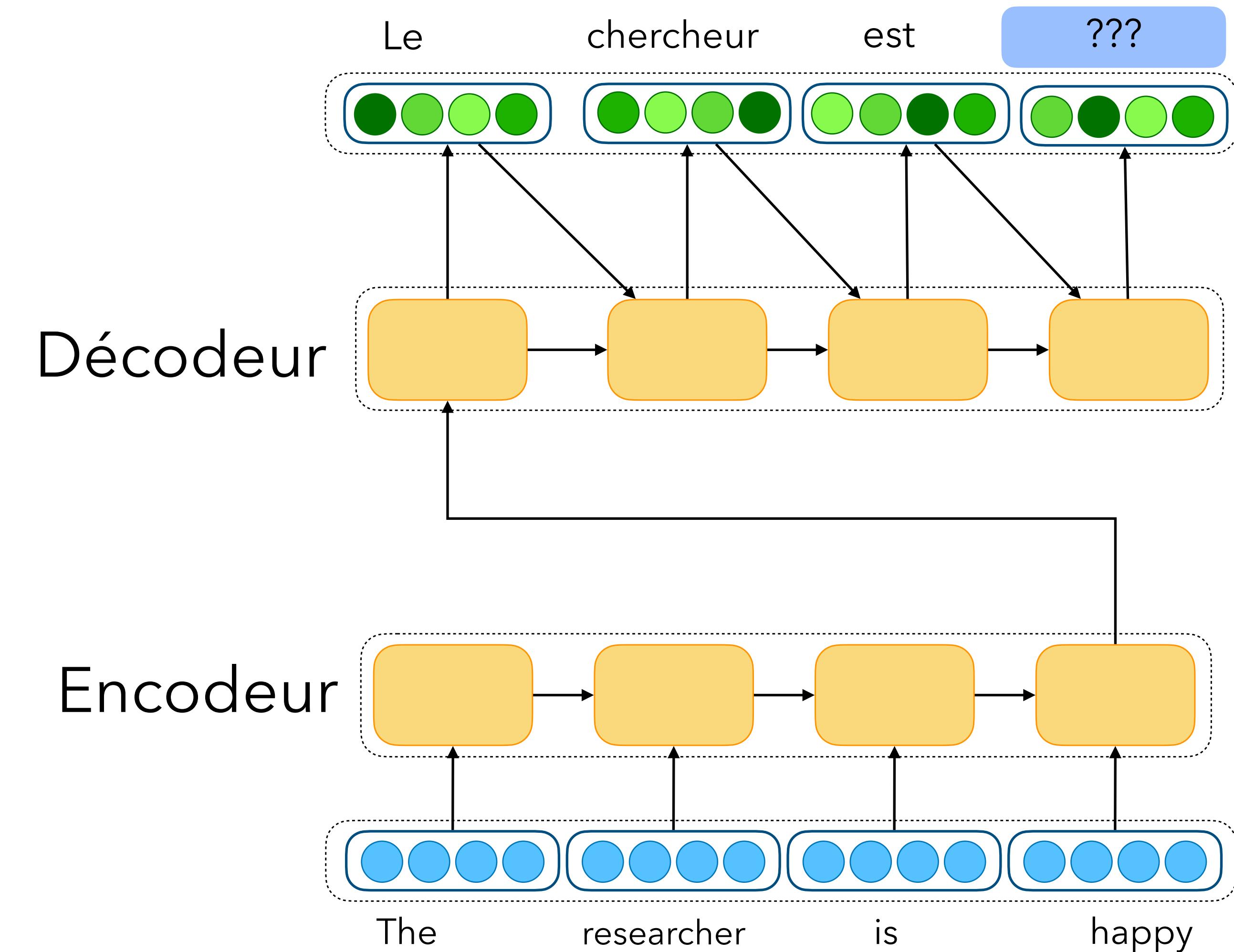
## 2. Partie technique

## 2. Partie technique

Rappel sur les *word embeddings*

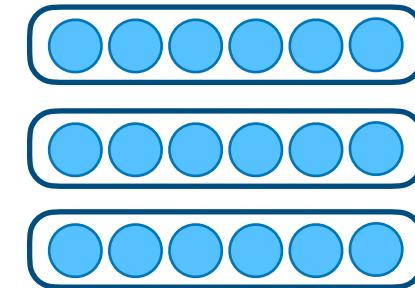
# Représentation des unités d'entrée

- Chaque mot d'entrée et de sortie est représenté par son *word embedding*
- *Word embedding* = représentation vectorielle (numérique) à haute dimension

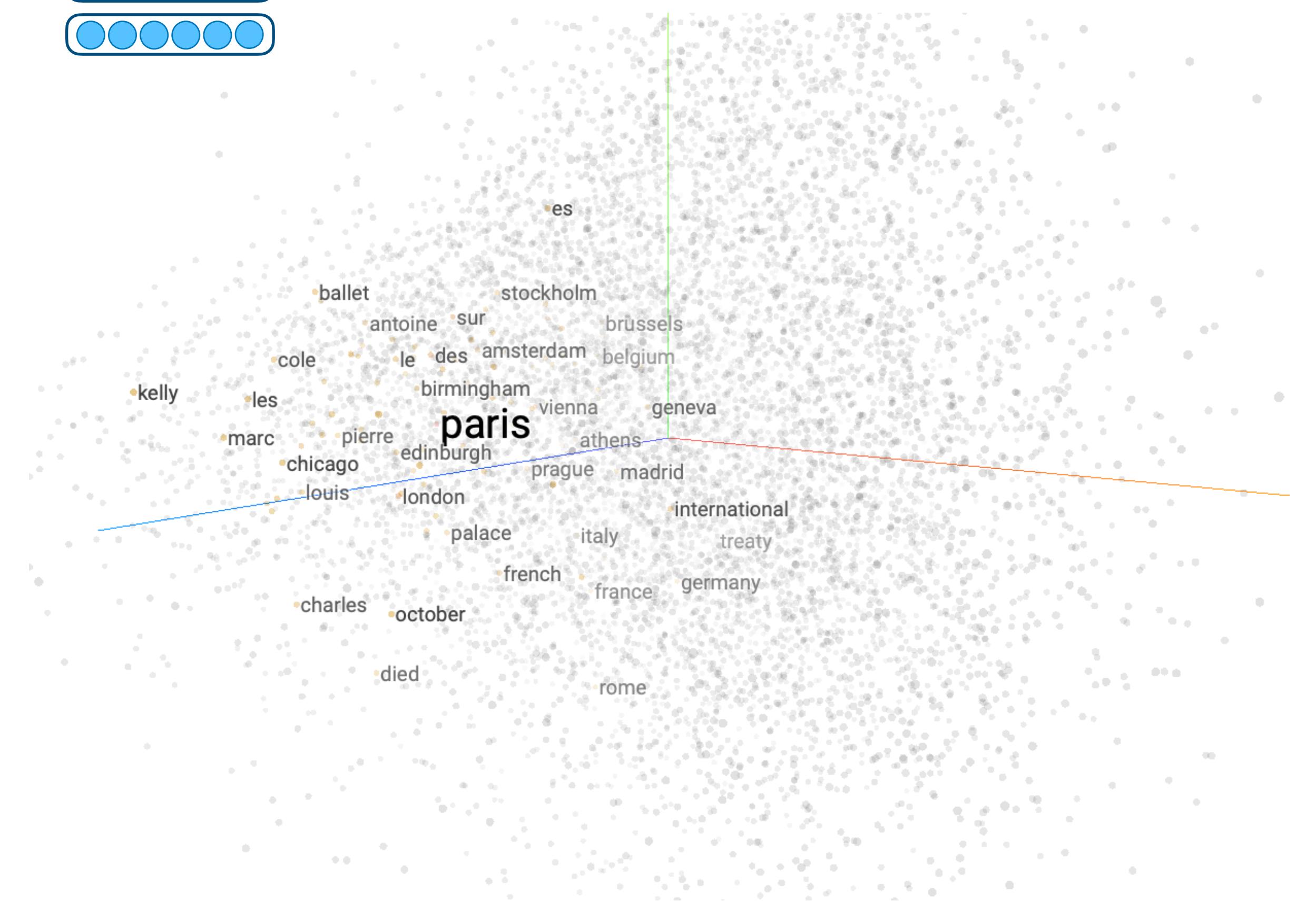


# Les word embeddings

rat	32	1	6	0	3	51
rats	21	11	0	4	2	44
lunch	2	35	0	0	12	2



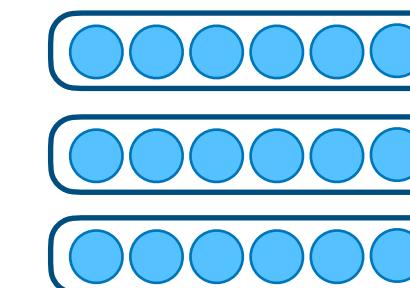
- chaque mot du vocabulaire est associé à un vecteur de nombre réels – la signification des ces nombres est abstraite
- Des mots similaires sont représentés par des *embeddings* similaires
- On peut visualiser cette similarité si on sélectionne (automatiquement) certaines dimensions et on représente les projections des embeddings dans ces dimensions :



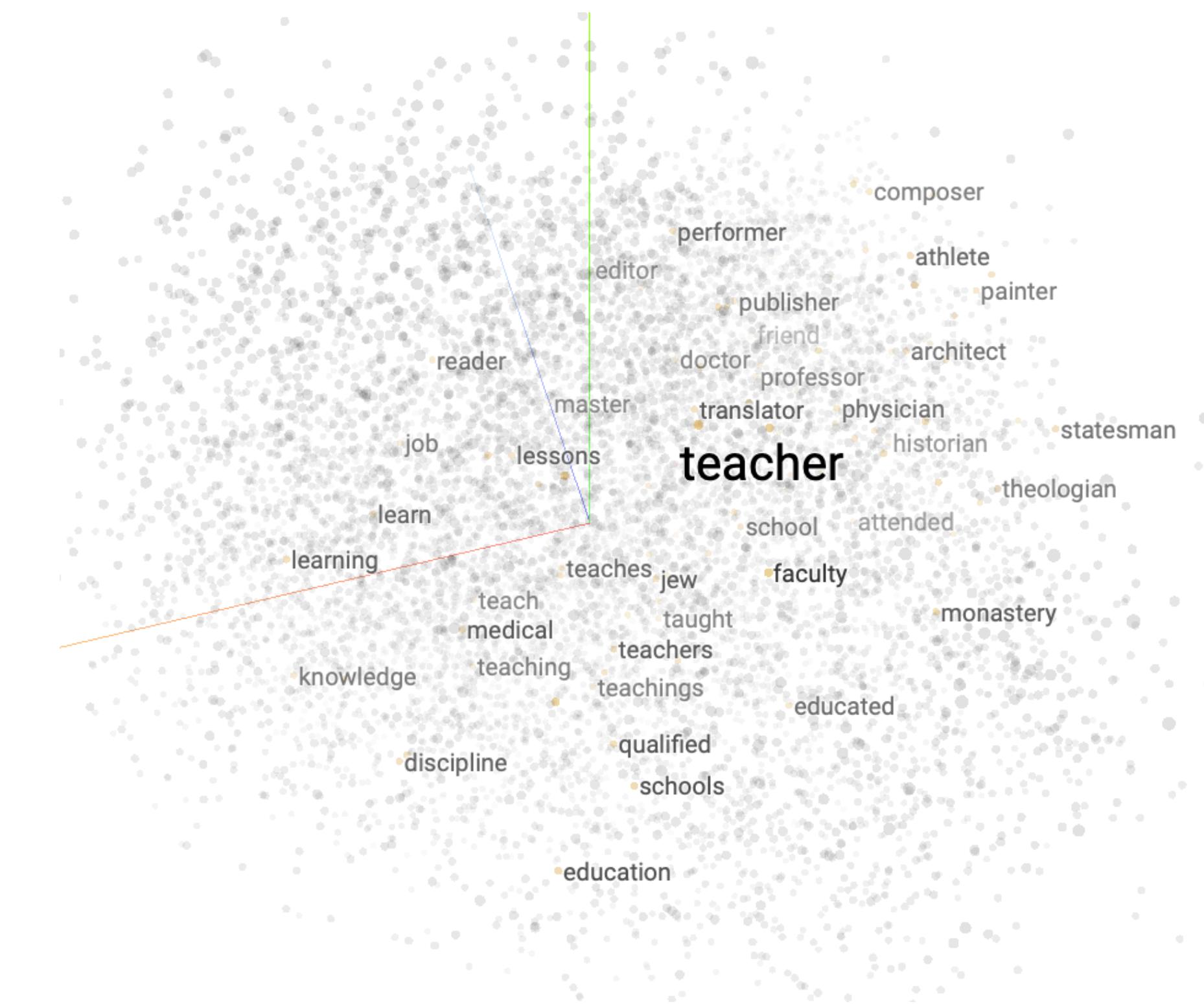
<https://projector.tensorflow.org>

# Les word embeddings

rat	32	1	6	0	3	51
rats	21	11	0	4	2	44
lunch	2	35	0	0	12	2



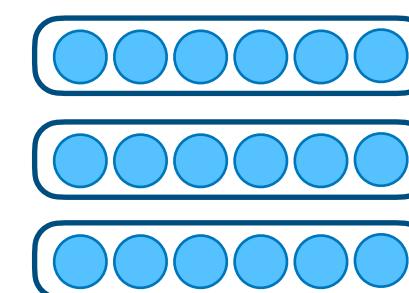
- chaque mot du vocabulaire est associé à un vecteur de nombre réels – la signification des ces nombres est abstraite
- Des mots similaires sont représentés par des *embeddings* similaires
- On peut visualiser cette similarité si on sélectionne (automatiquement) certaines dimensions et on représente les projections des embeddings dans ces dimensions :



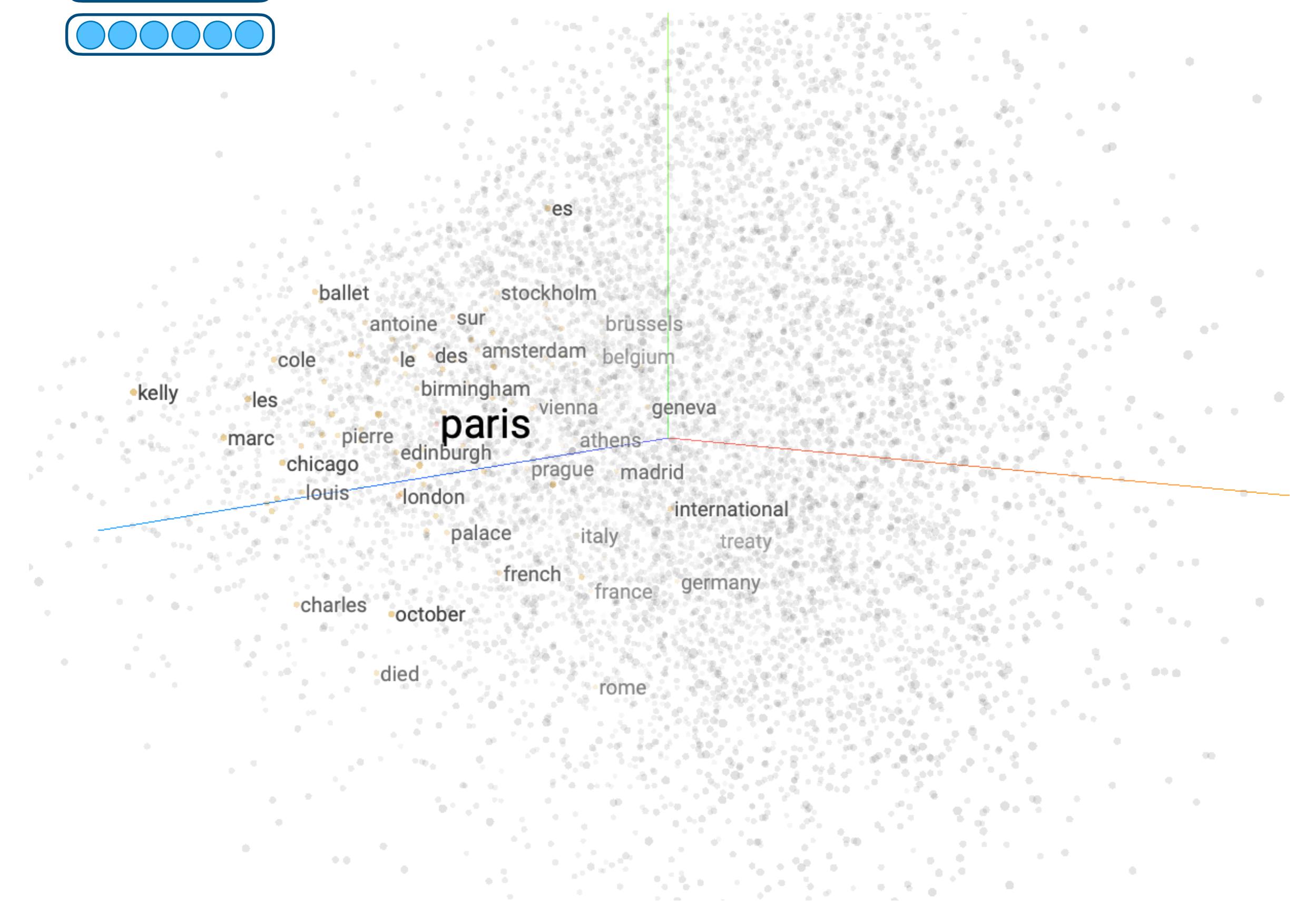
<https://projector.tensorflow.org>

# Les word embeddings

<i>rat</i>	32	1	6	0	3	51
<i>rats</i>	21	11	0	4	2	44
<i>lunch</i>	2	35	0	0	12	2



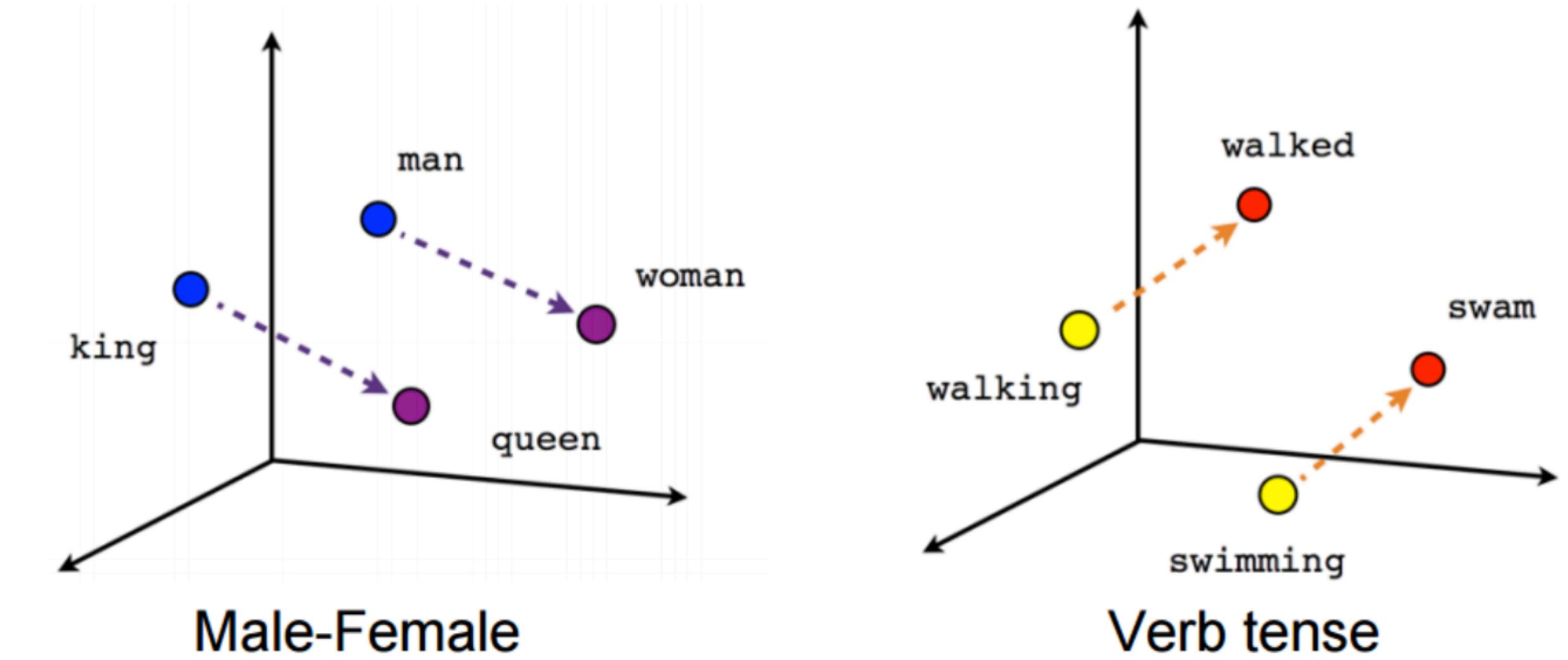
- chaque mot du vocabulaire est associé à un vecteur de nombre réels – la signification des ces nombres est abstraite
  - Des mots similaires sont représentés par des *embeddings* similaires
  - On peut visualiser cette similarité si on sélectionne (automatiquement) certaines dimensions et on représente les projections des embeddings dans ces dimensions :



<https://projector.tensorflow.org>

# Transformation de word embeddings

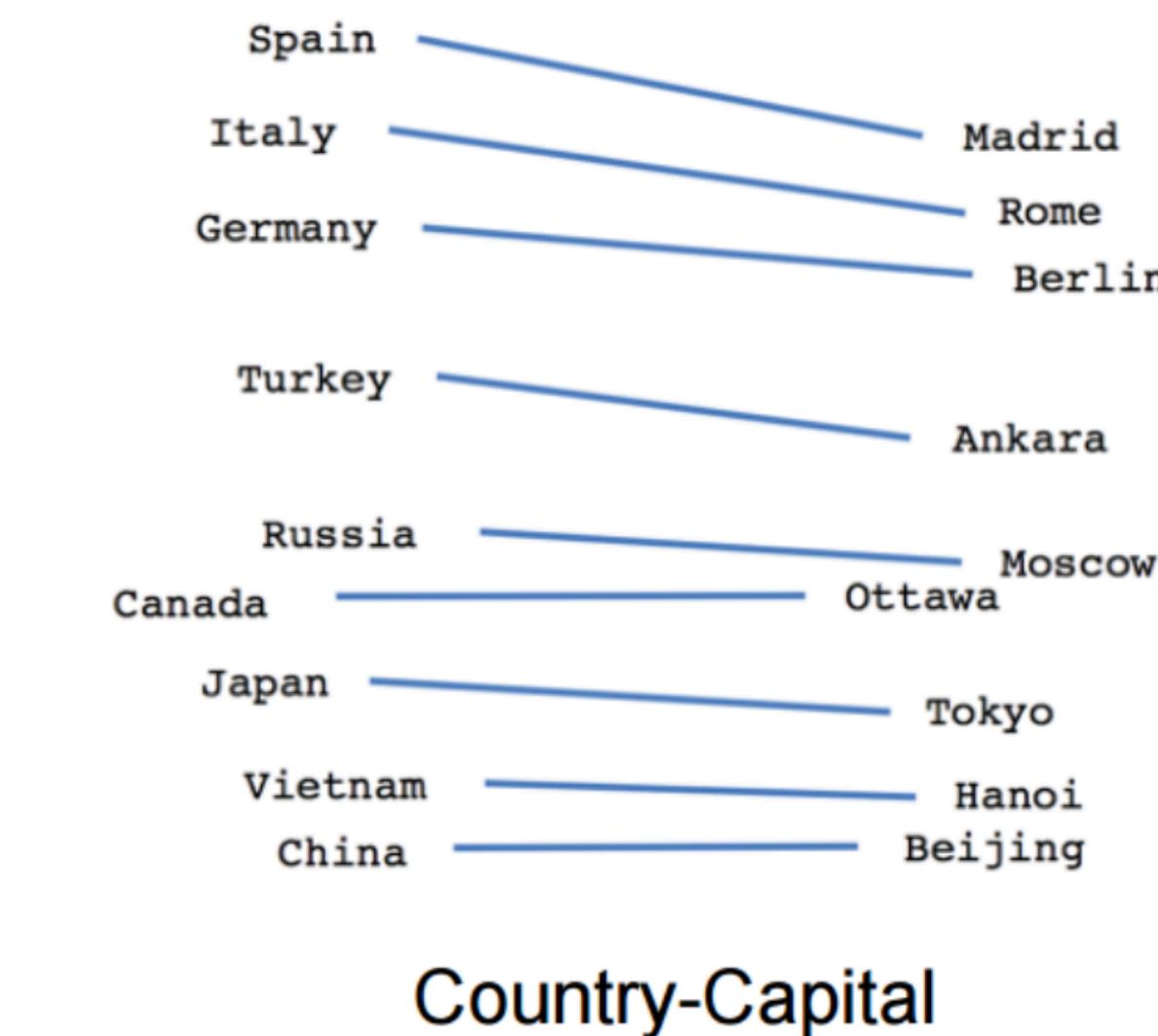
- Les embeddings sont des objets mathématiques et on peut appliquer des transformations numériques (par ex. : addition, soustraction, multiplication, division, etc.)



- Il est possible d'apprendre des transformations de différentes natures, qui représentent des relations entre mots :

- $king - man + woman \approx queen$

- $walking - walked + swam \approx swimming$

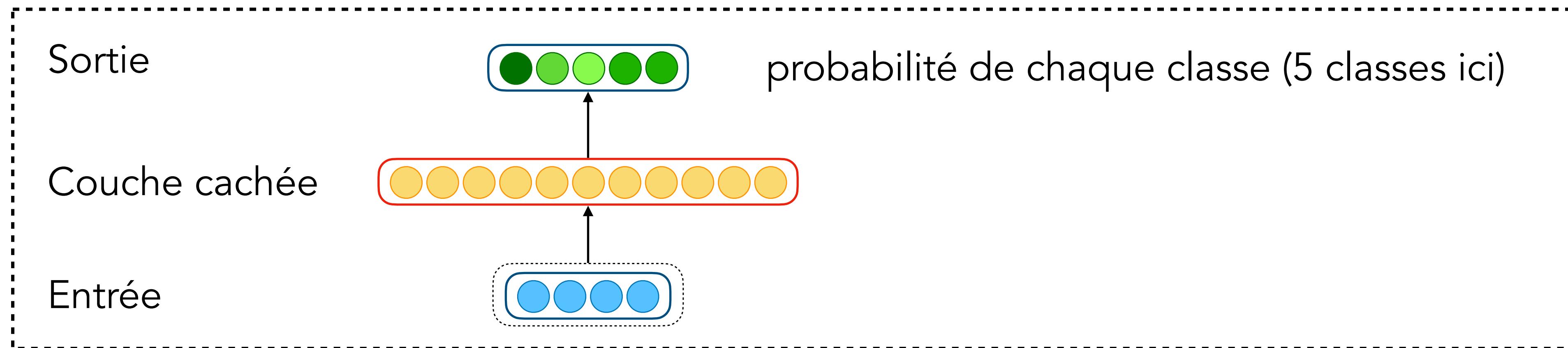


## 2. Partie technique

Explication des architectures de traduction

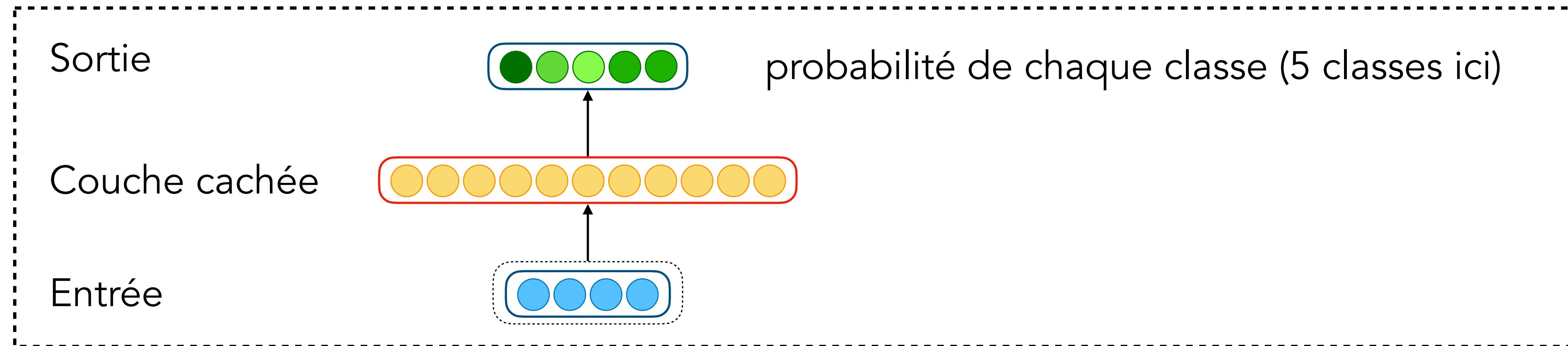
# Réseau de neurones simple

Un réseau de neurone simple : une fonction de transformation complexe composée de couches successives de transformations plus simples



# Réseau de neurones simple

Un réseau de neurone simple : une fonction de transformation complexe composée de couches successives de transformations plus simples



La couche cachée représente une transformation des unités d'entrée:

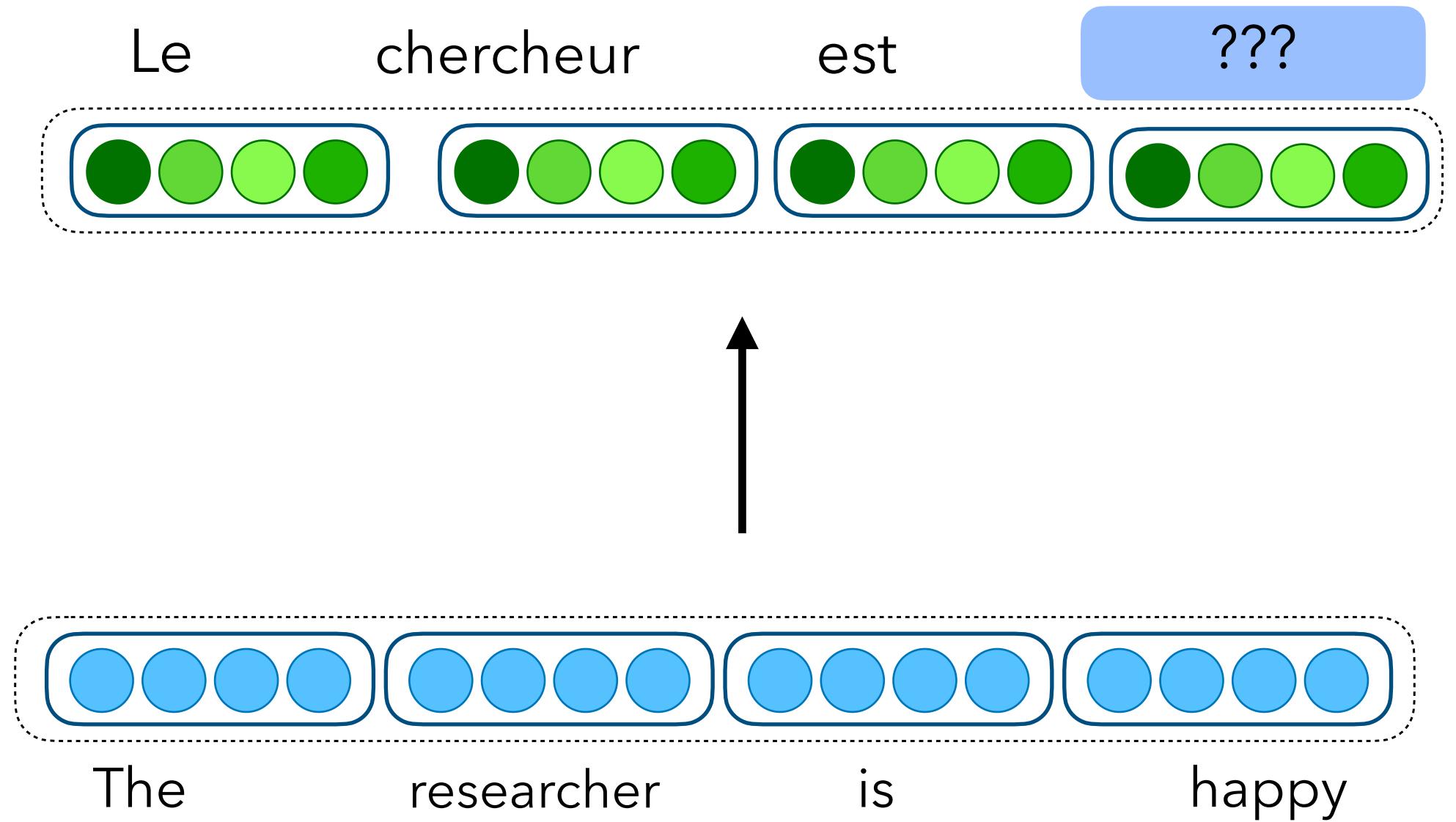
- Une transformation linéaire (multiplier les valeurs par des valeurs apprises, appelées paramètres)
- Une transformation non linéaire (transformer les représentations autrement)

Couche de sortie

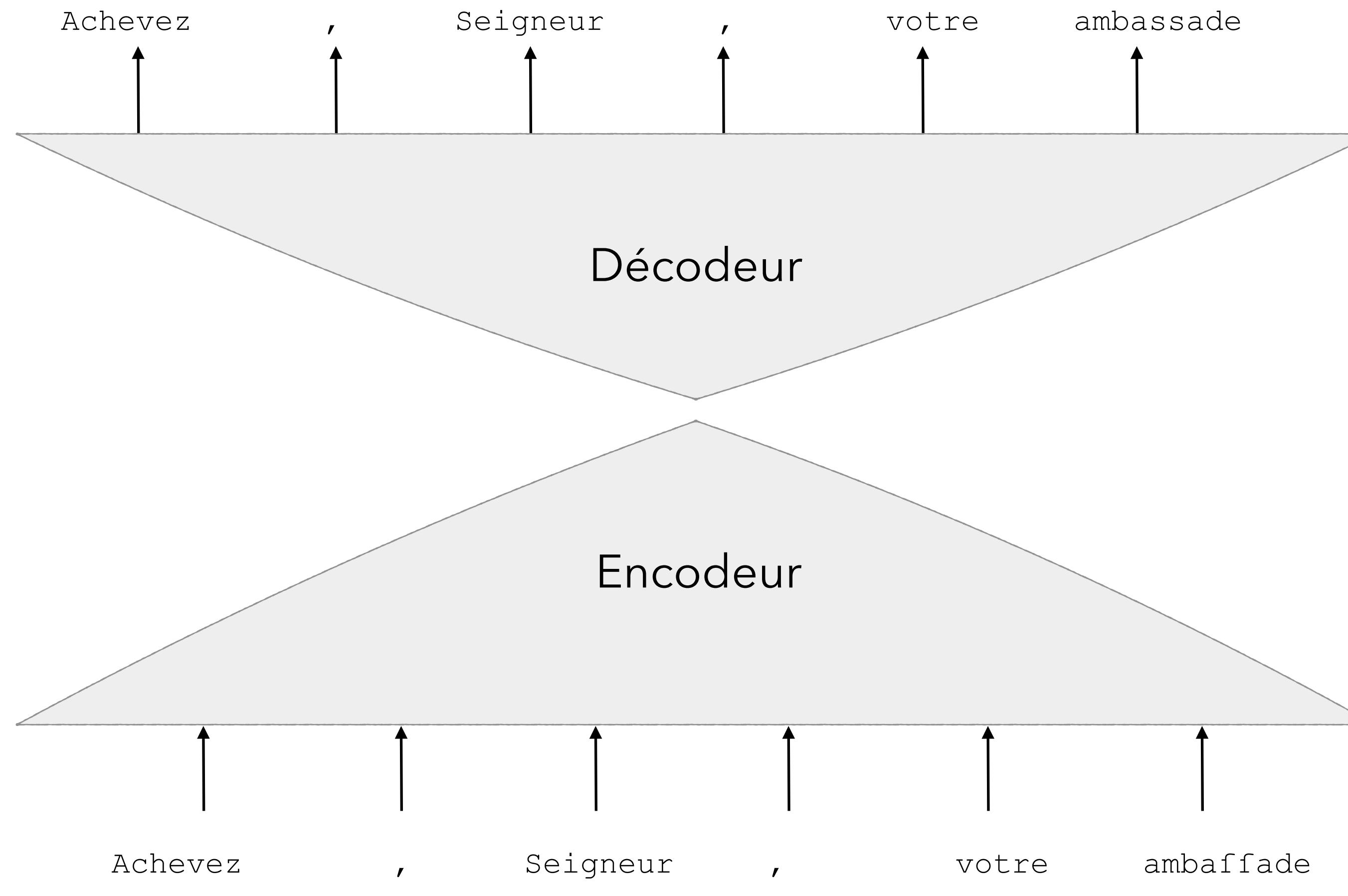
- Une transformation linéaire
- Une fonction d'activation finale (ex. : la fonction softmax, qui donne une distribution de probabilité sur les classes possibles)

# Traduction automatique neuronale

- Une séquence de mots en entrée
- On veut produire une séquence de mots en sortie (pas forcément de la même longueur que le texte d'entrée)
- On veut pouvoir transformer le texte d'entrée en le texte de sortie
  - Chaque mot d'entrée = son *word embedding* (en bleu)
  - En sortie nous avons une distribution de probabilité sur les classes possibles = les mots du vocabulaire (en vert),
  - On cherche la séquence de mots qui donne la probabilité maximale étant donné le texte en entrée

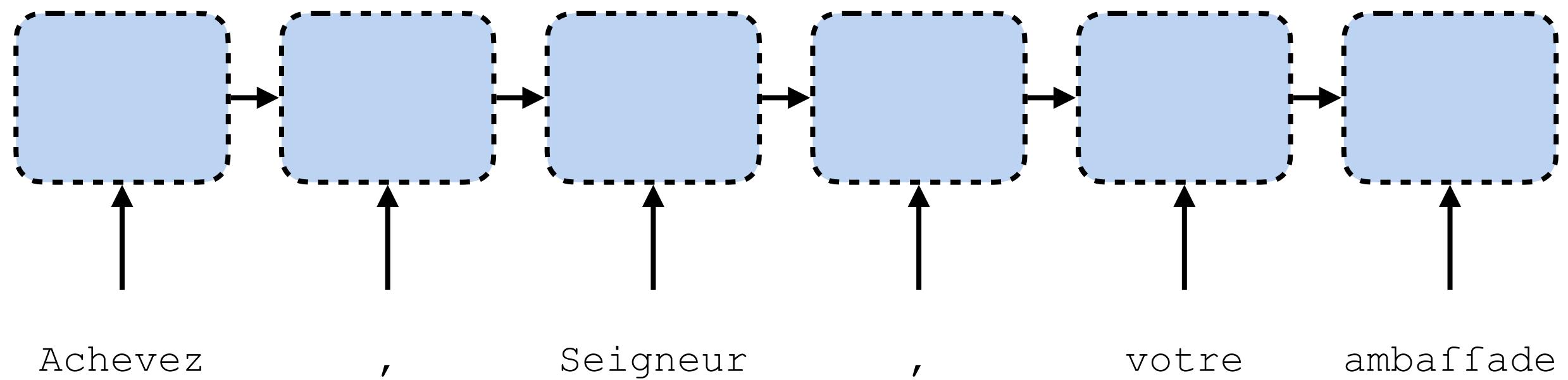


# Modèle sequence-to-sequence de type encodeur-décodeur



1. Les mots de la phrase d'origine
2. Transformation numérique des mots de la phrase d'origine
3. Produire mot par mot la phrase normalisée
4. Les mots choisis (qui donnent la probabilité maximale selon le modèle)

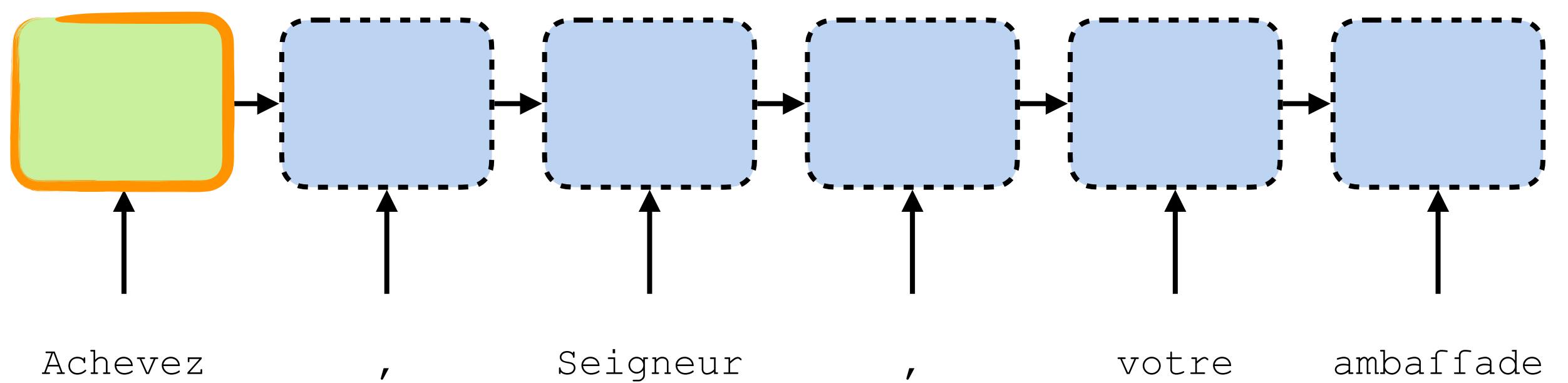
# Modèle encodeur-décodeur récurrente (RNN)



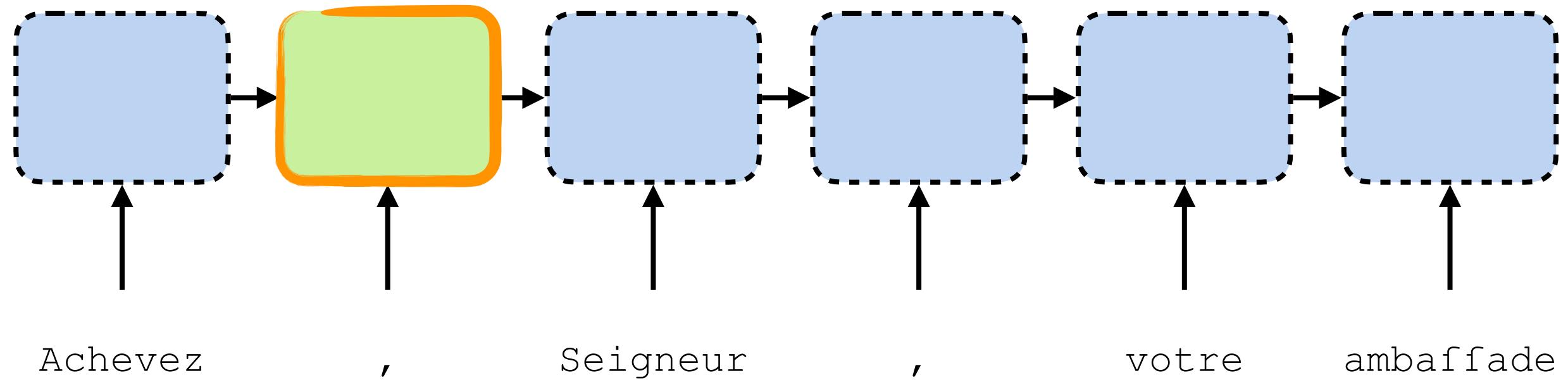
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)

- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

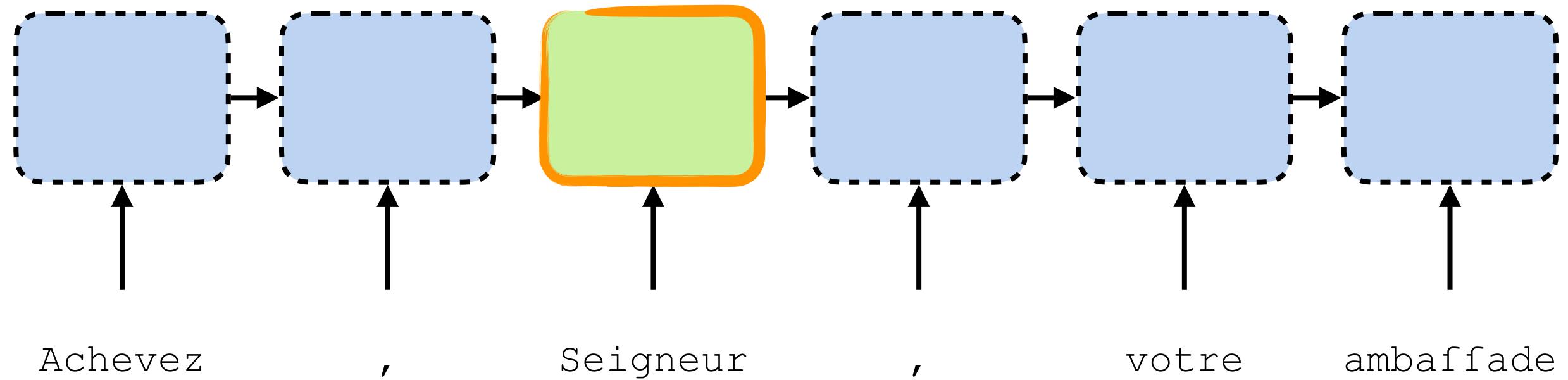


# Modèle encodeur-décodeur récurrente (RNN)



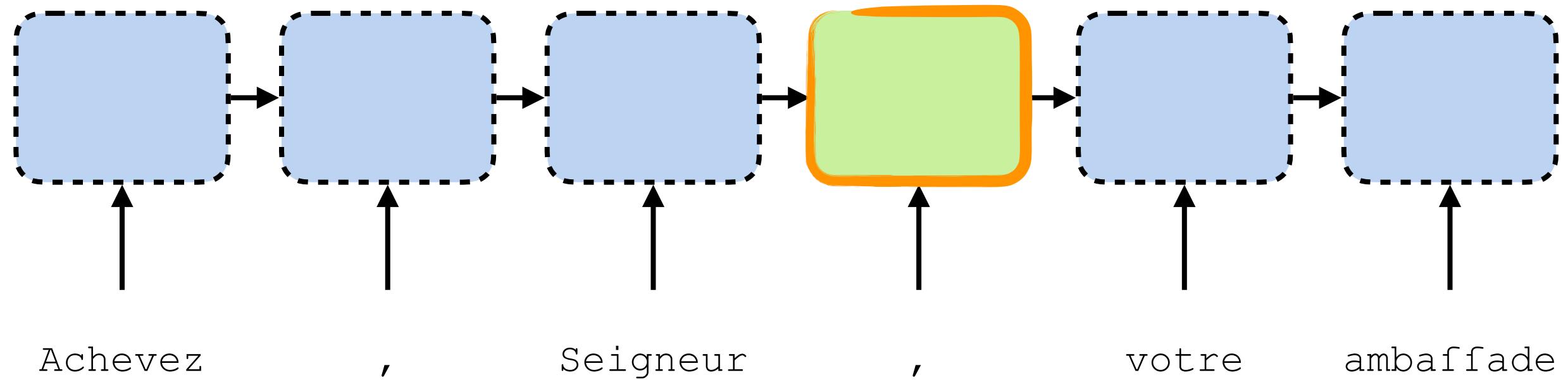
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



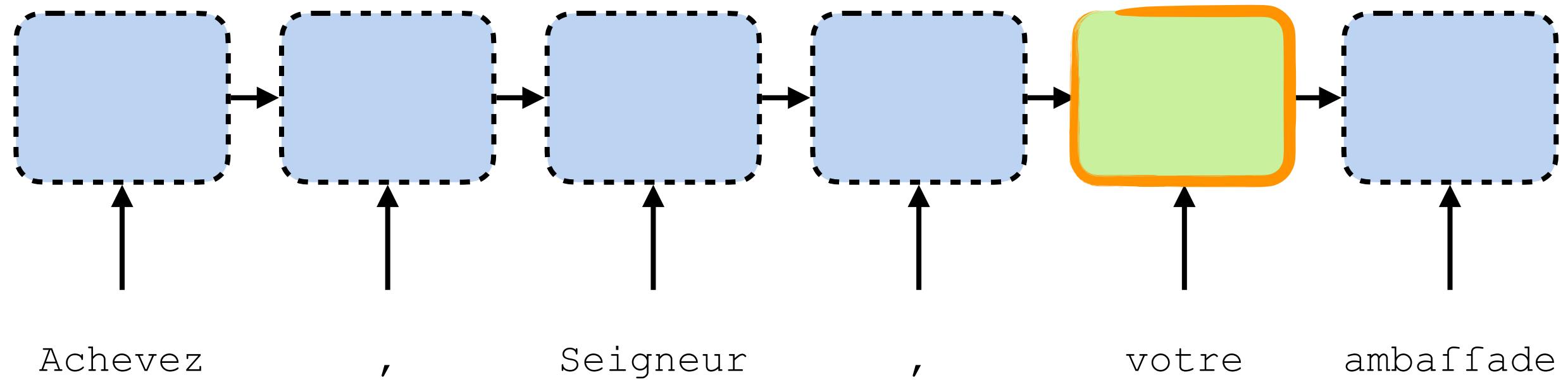
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



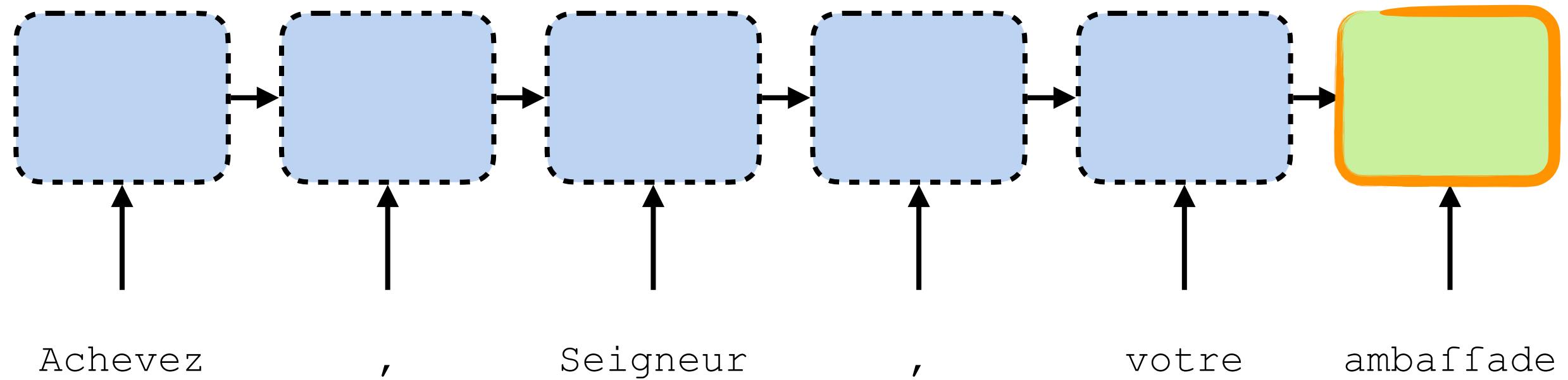
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



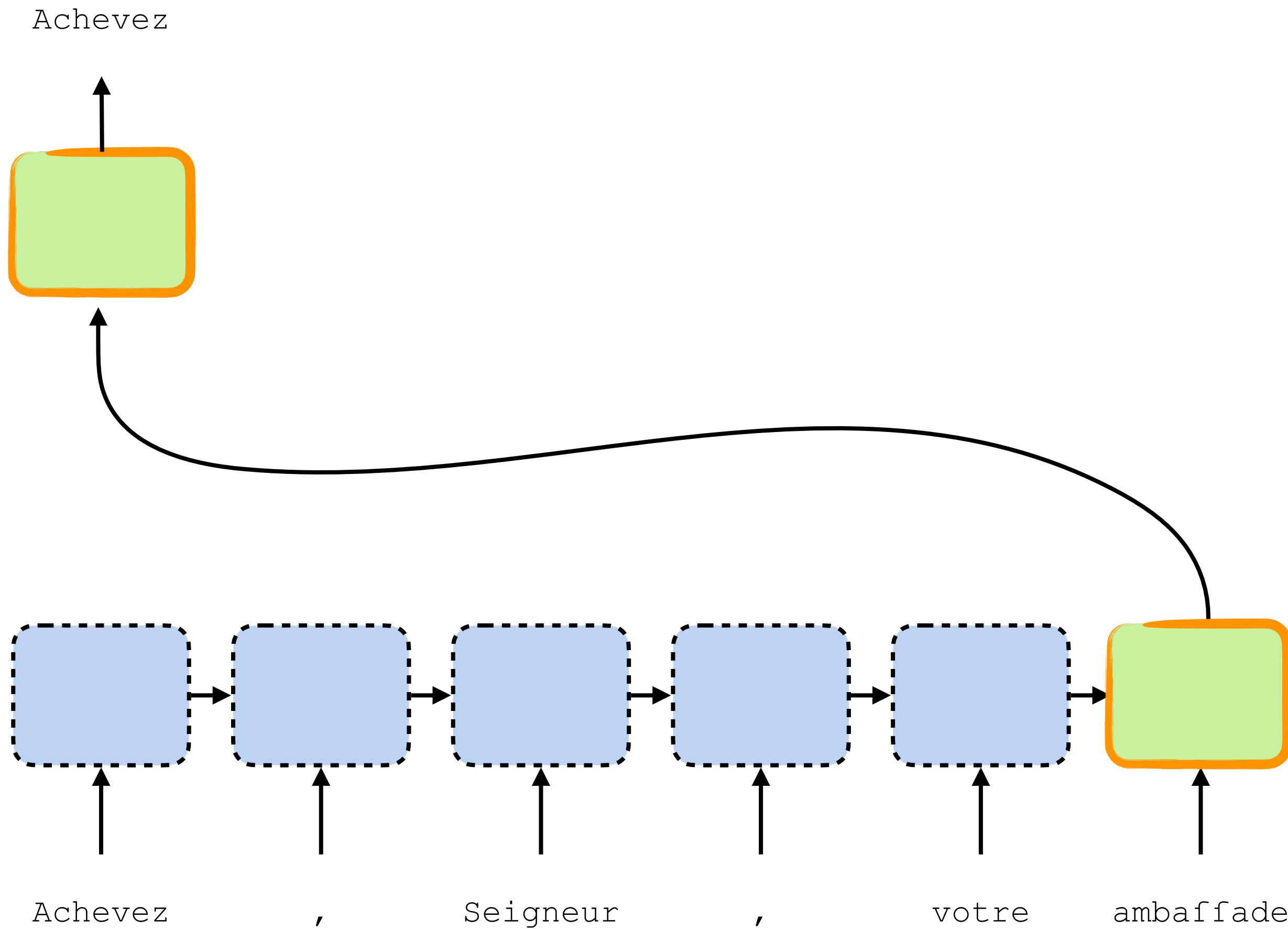
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



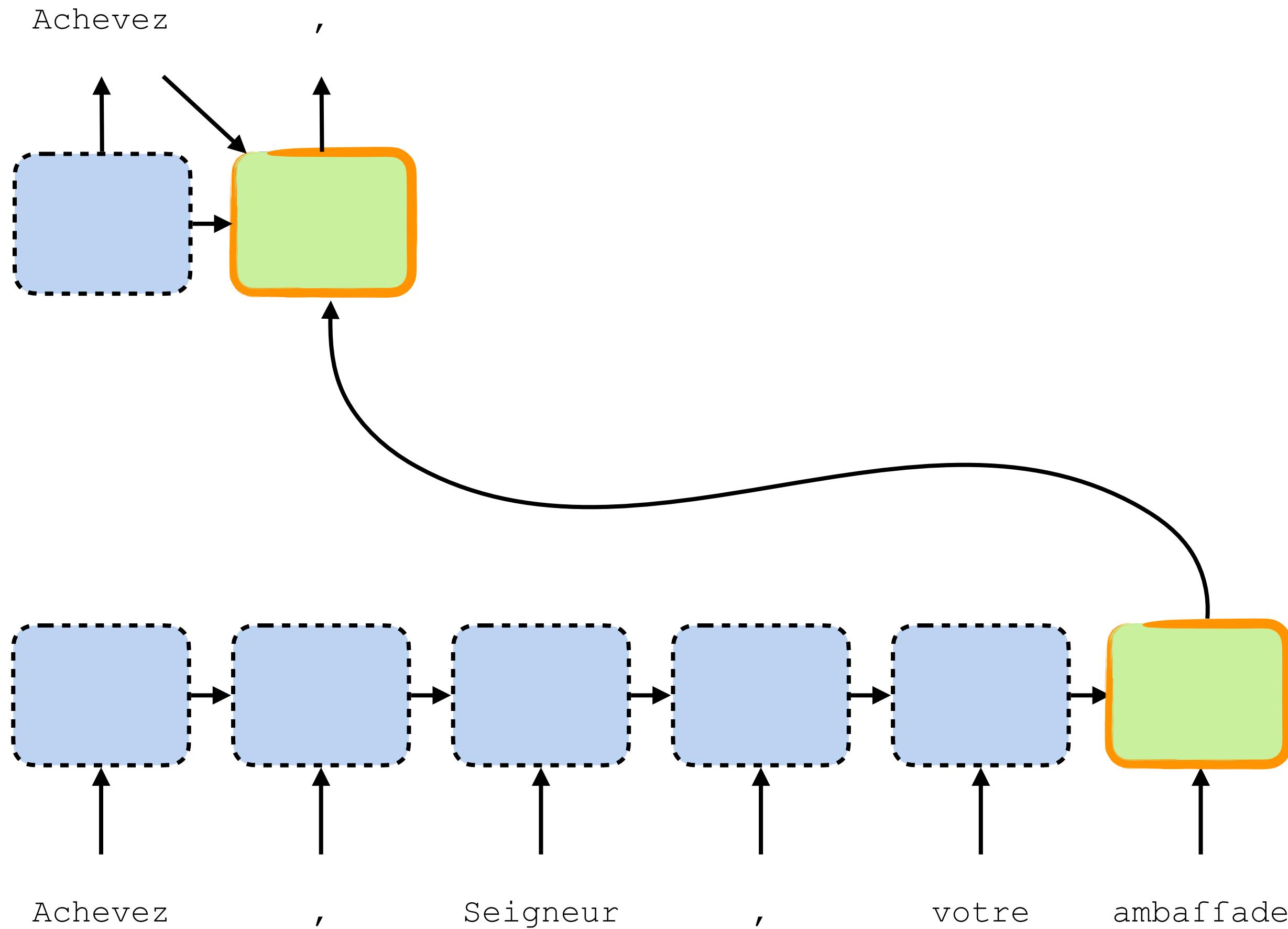
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



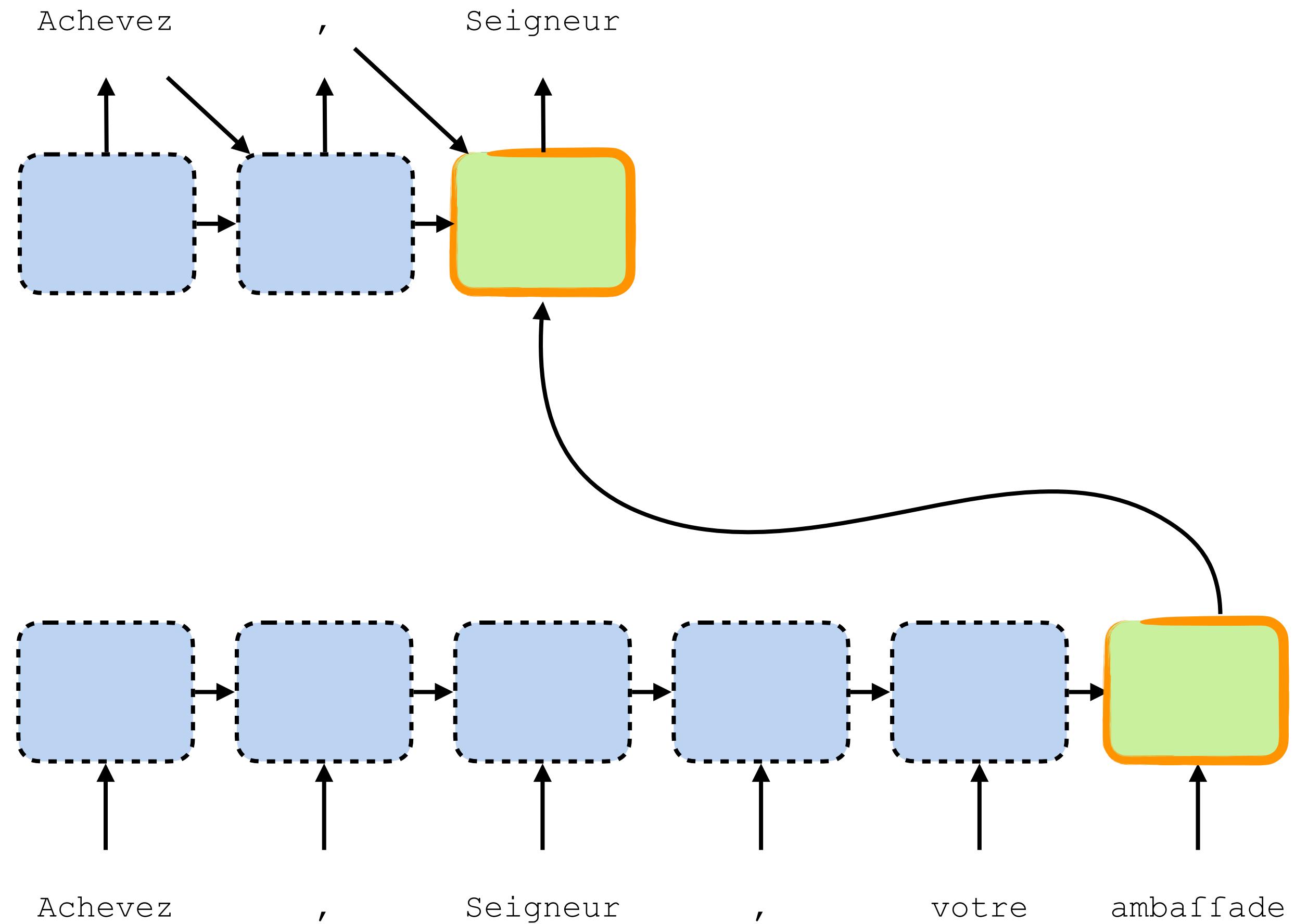
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



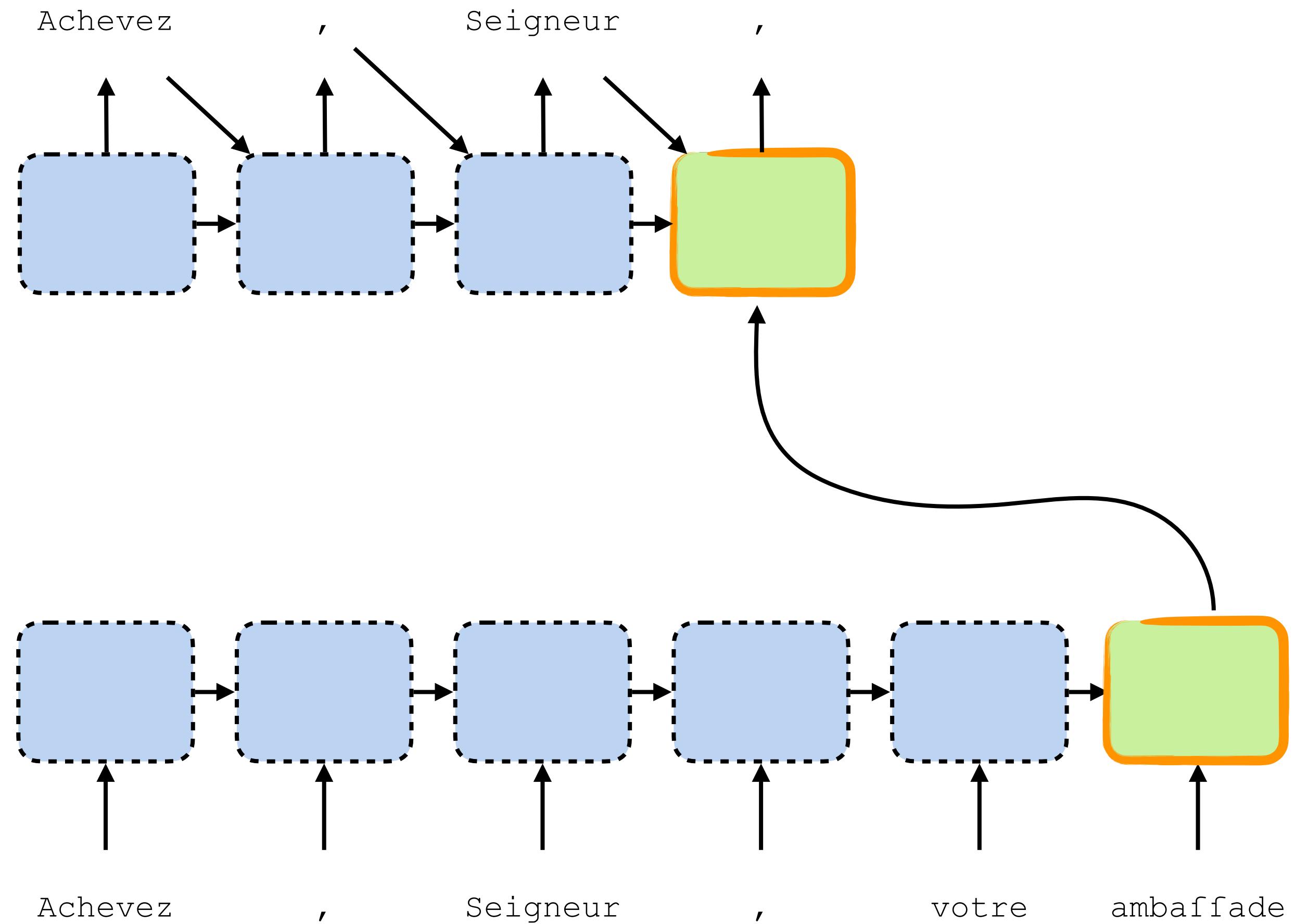
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



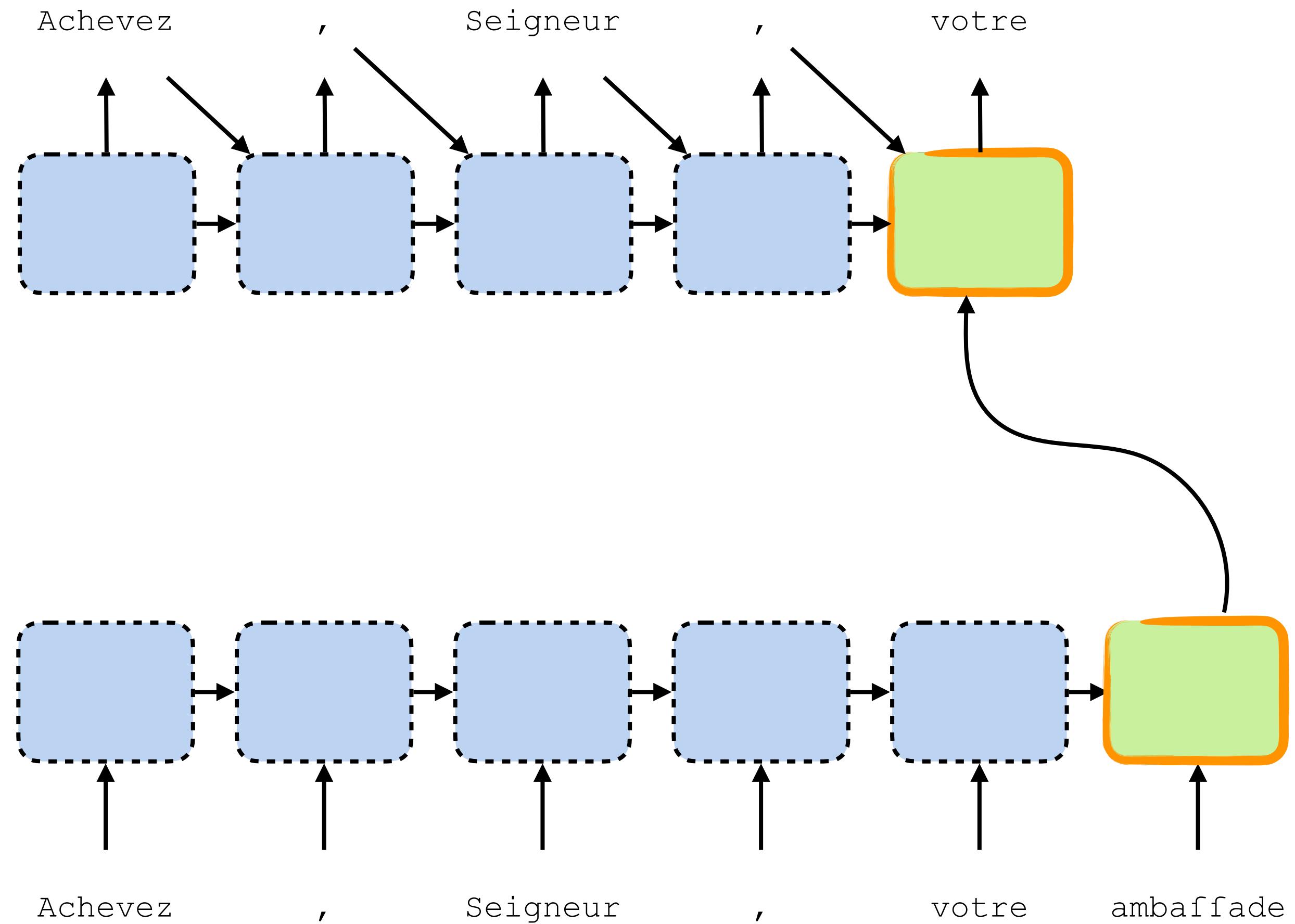
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



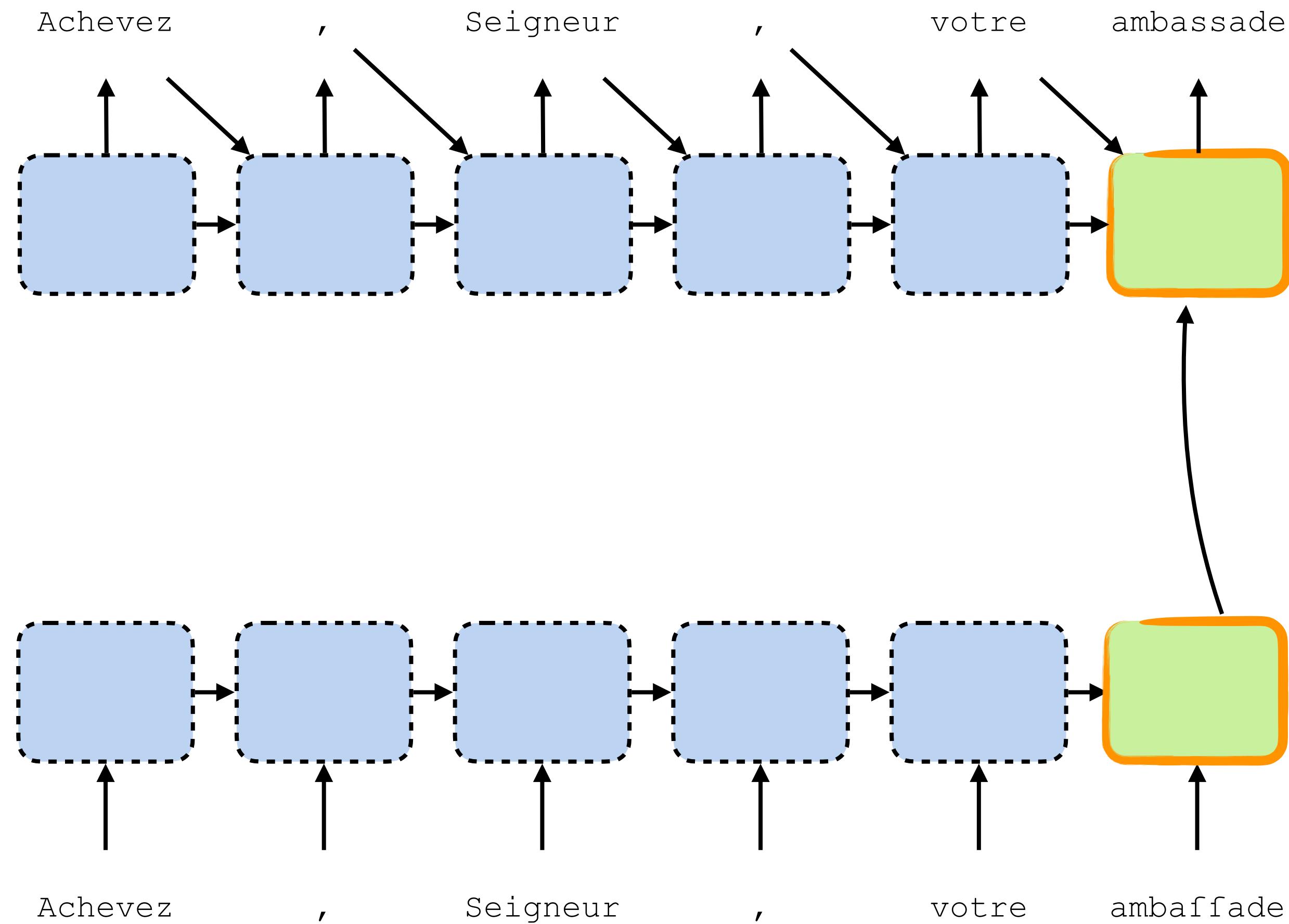
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



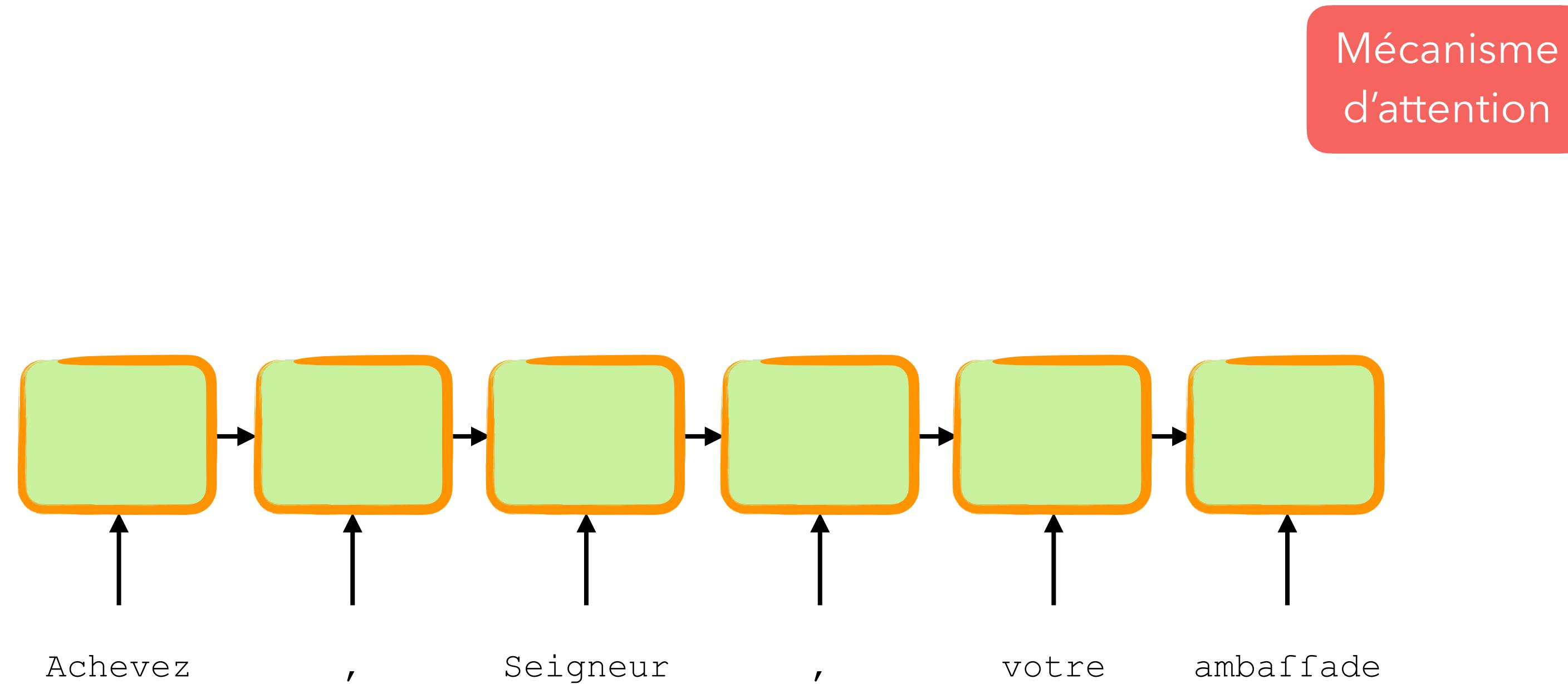
- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

# Modèle encodeur-décodeur récurrente (RNN)



- Première modèle de traduction
- Modèle récurrent : encoder de façon récurrente les mots du texte d'entrée
- Utiliser la représentation finale pour ensuite produire de façon récurrente les mots de sortie

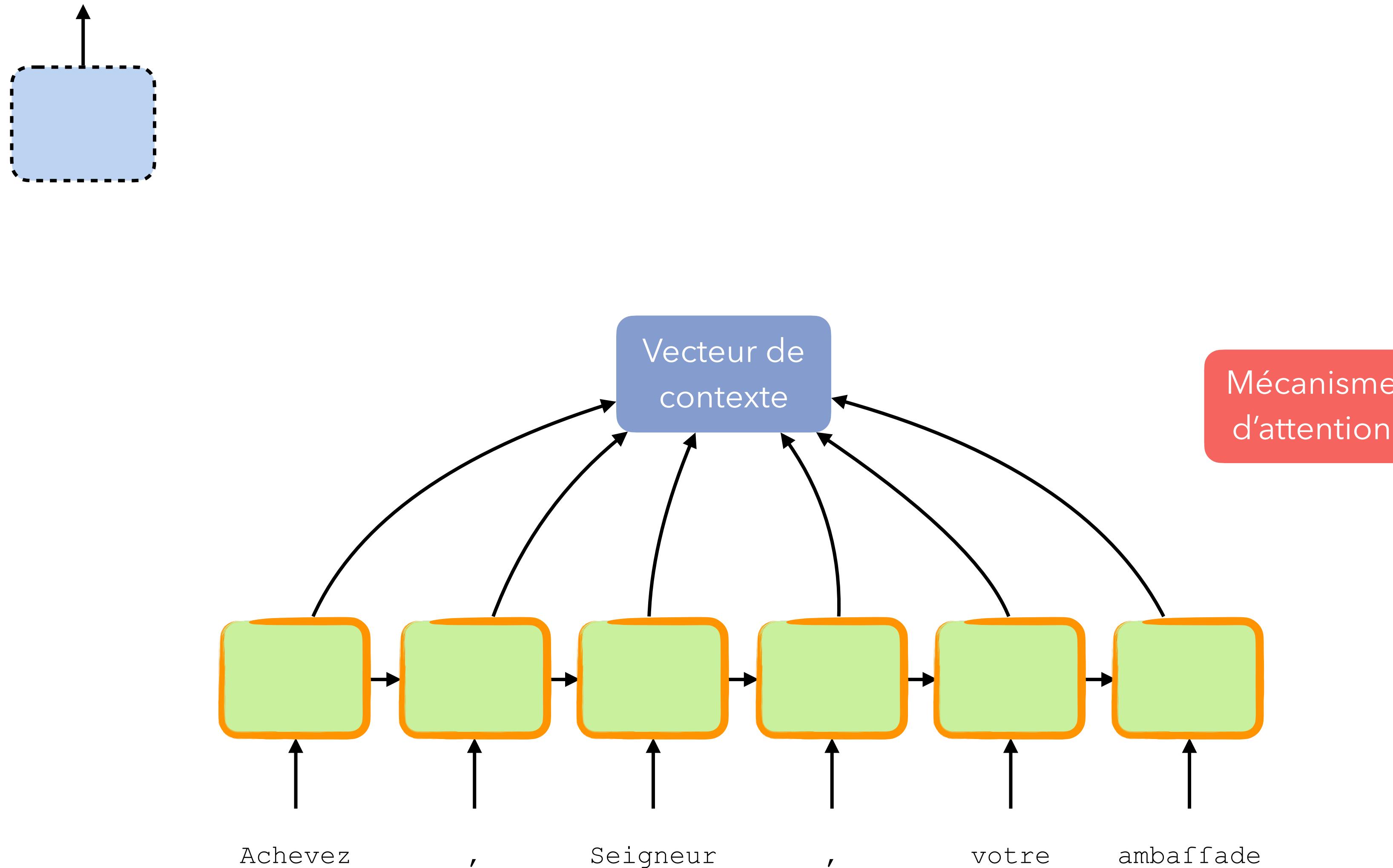
# Amélioration : mécanisme d'attention



- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

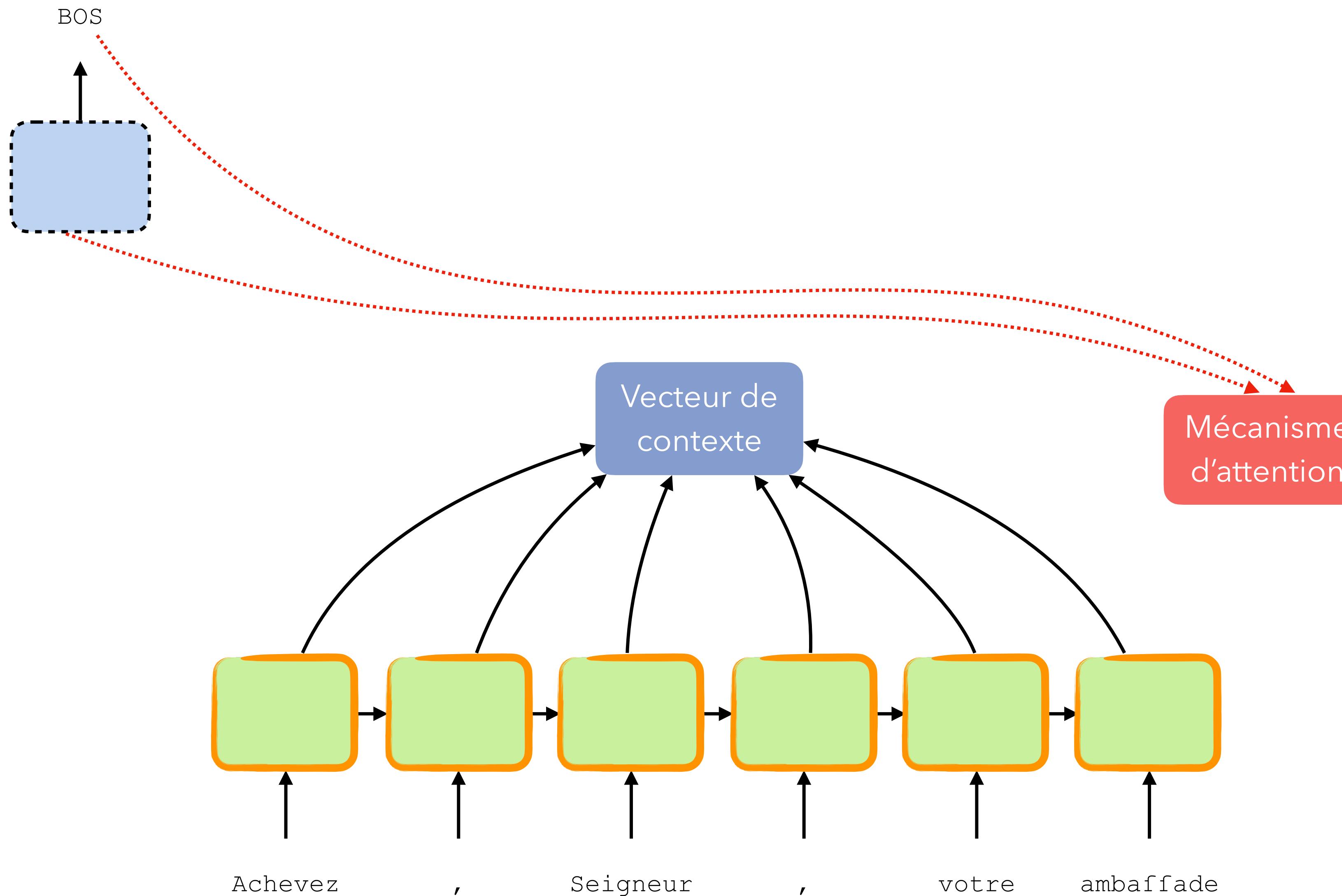
# Amélioration : mécanisme d'attention

BOS



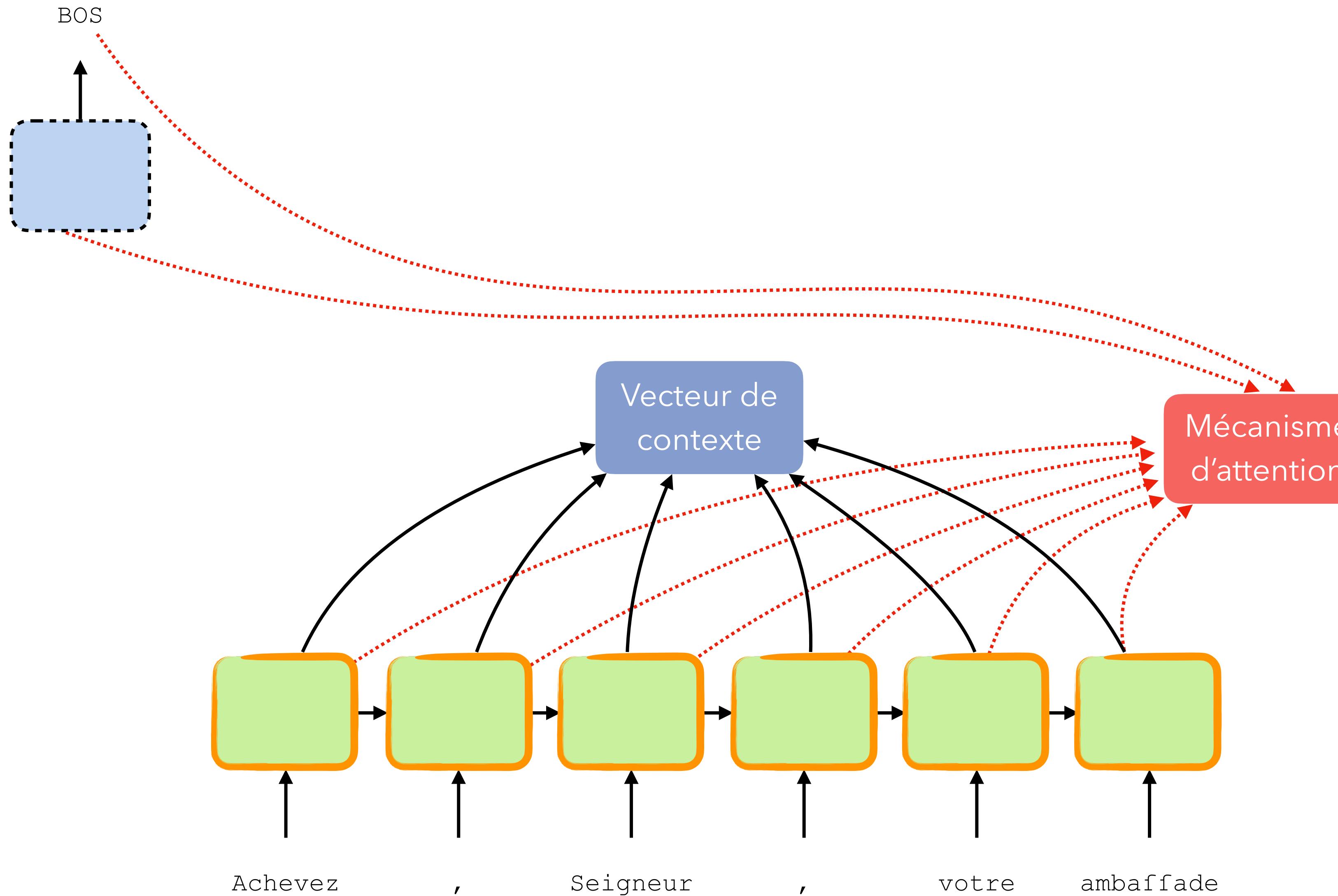
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

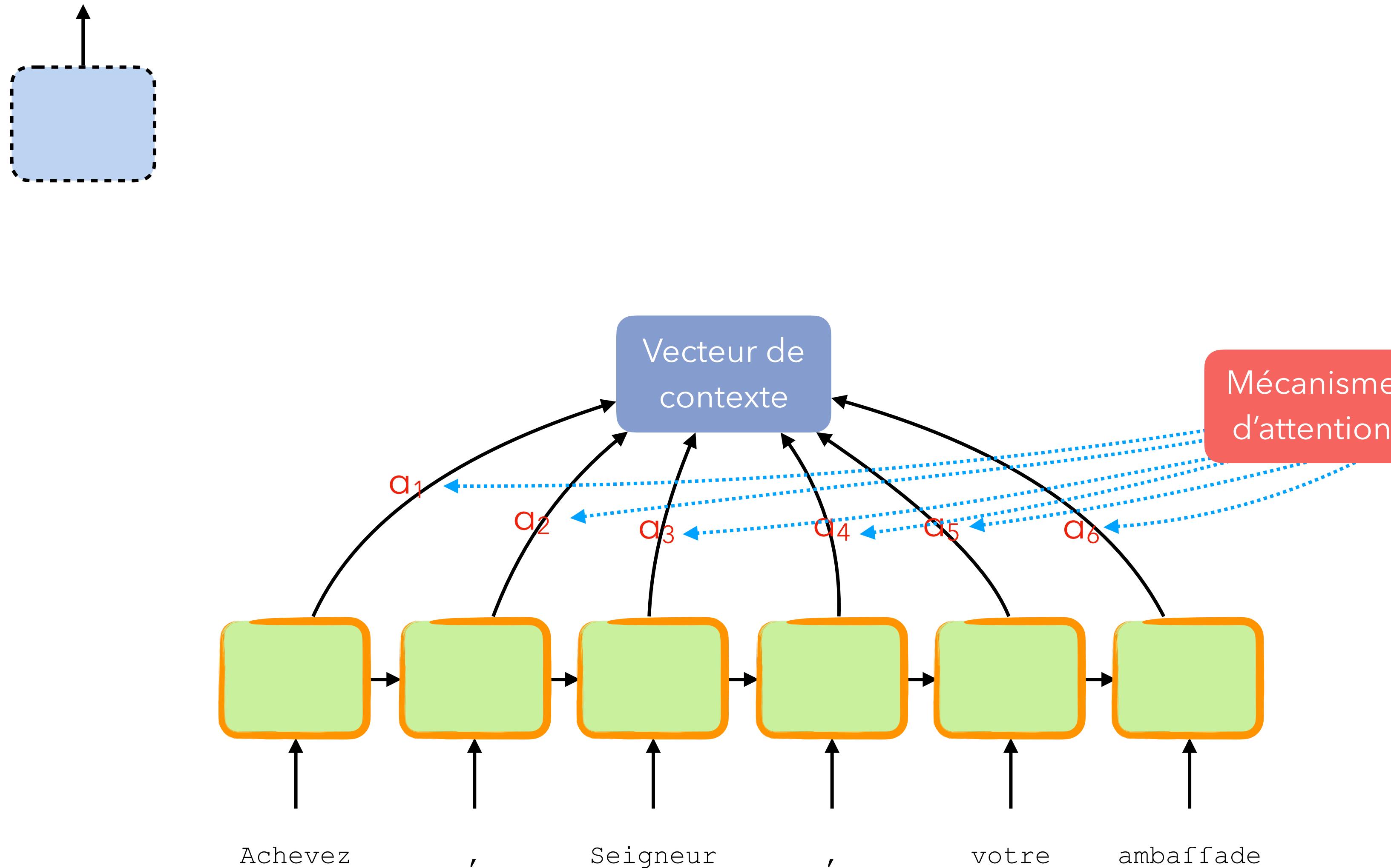
# Amélioration : mécanisme d'attention



- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

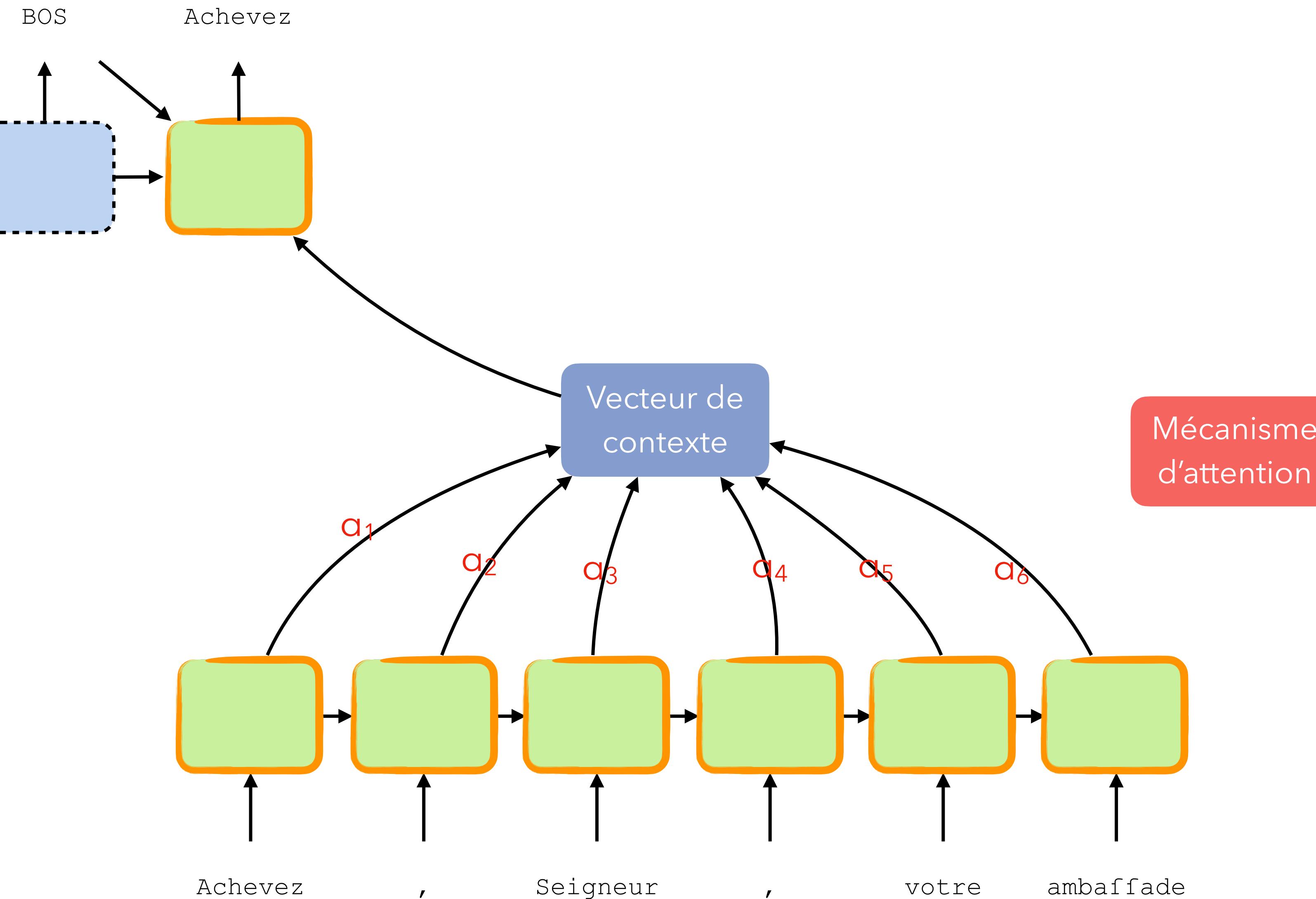
# Amélioration : mécanisme d'attention

BOS



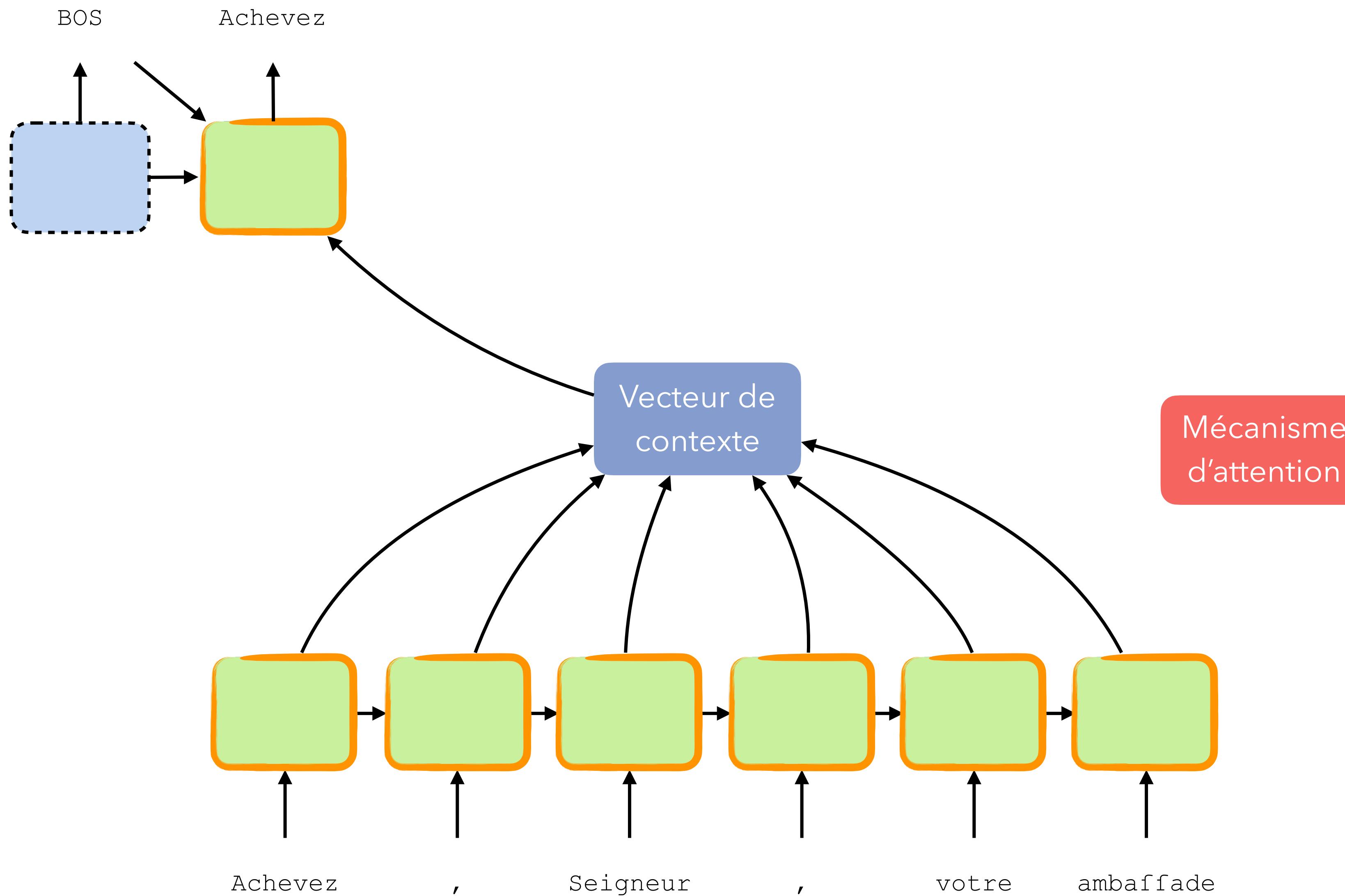
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



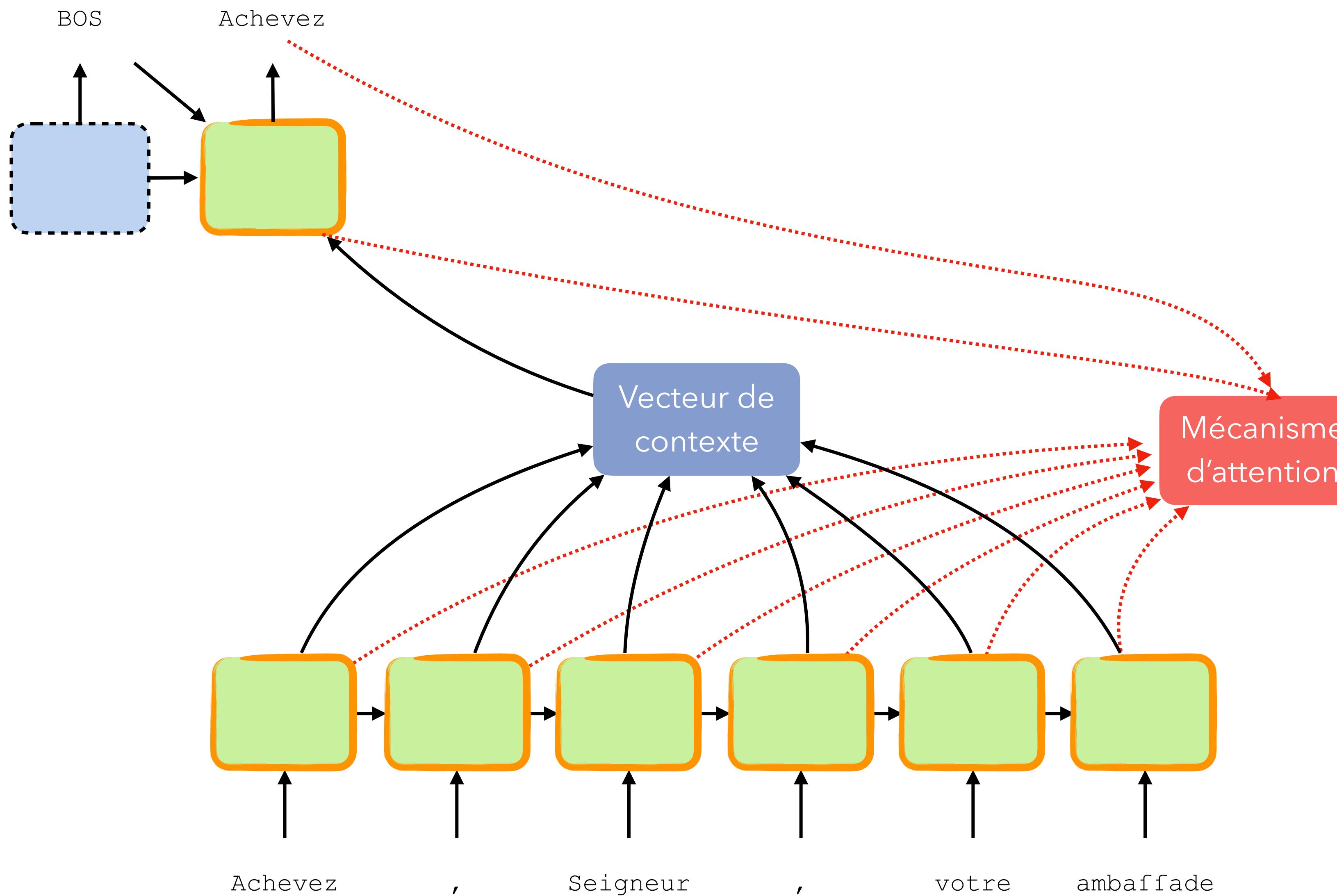
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



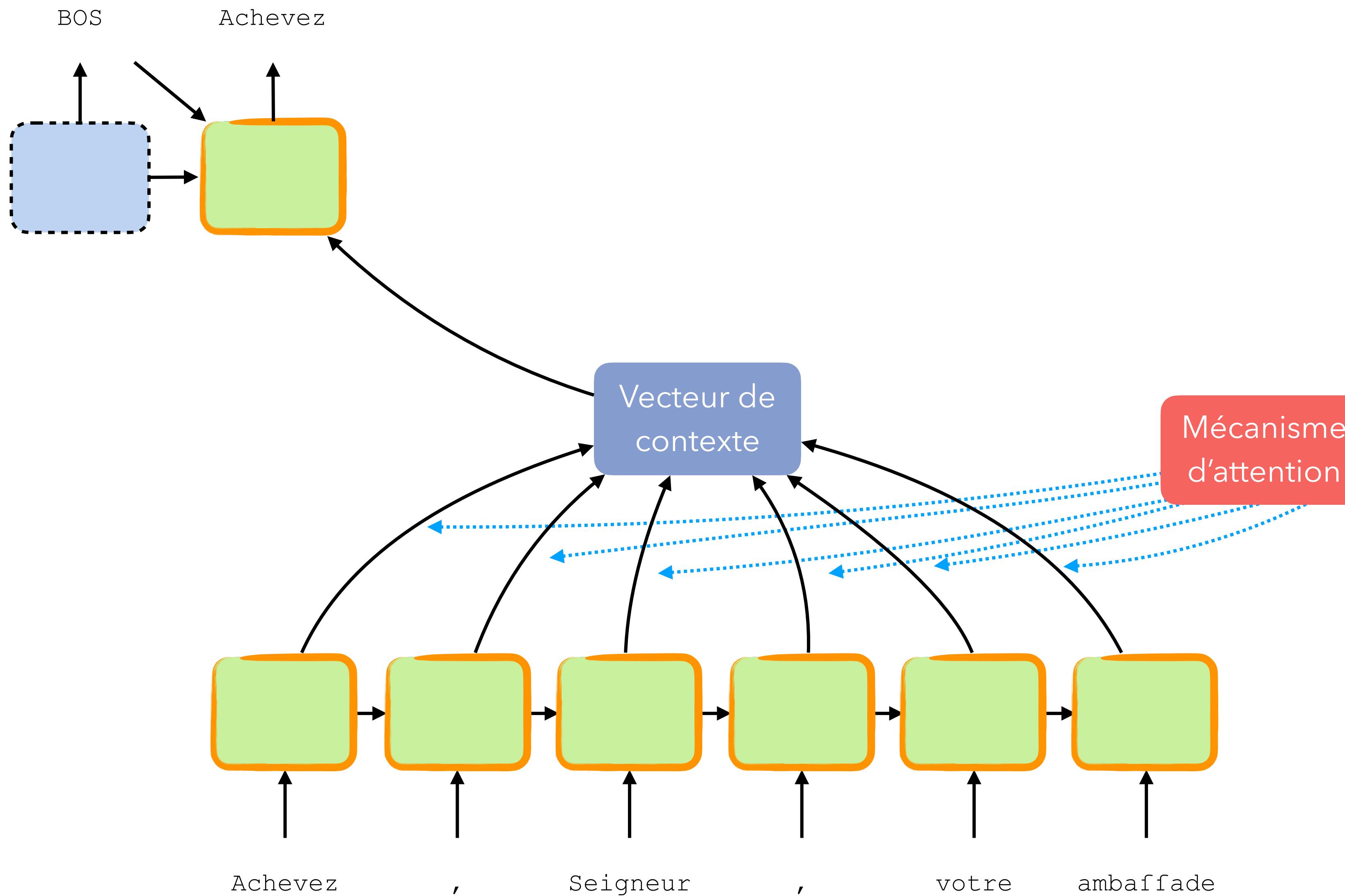
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



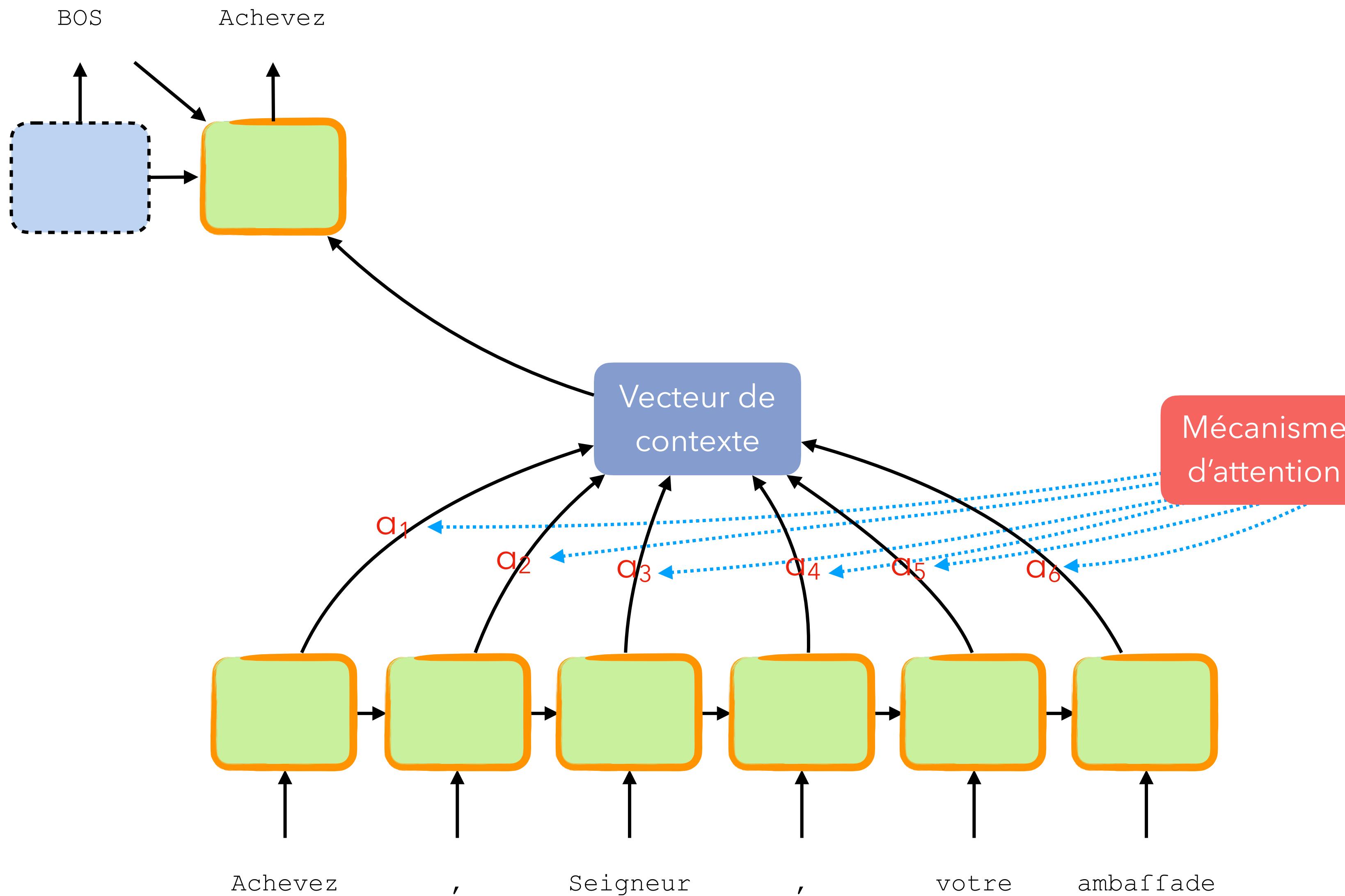
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



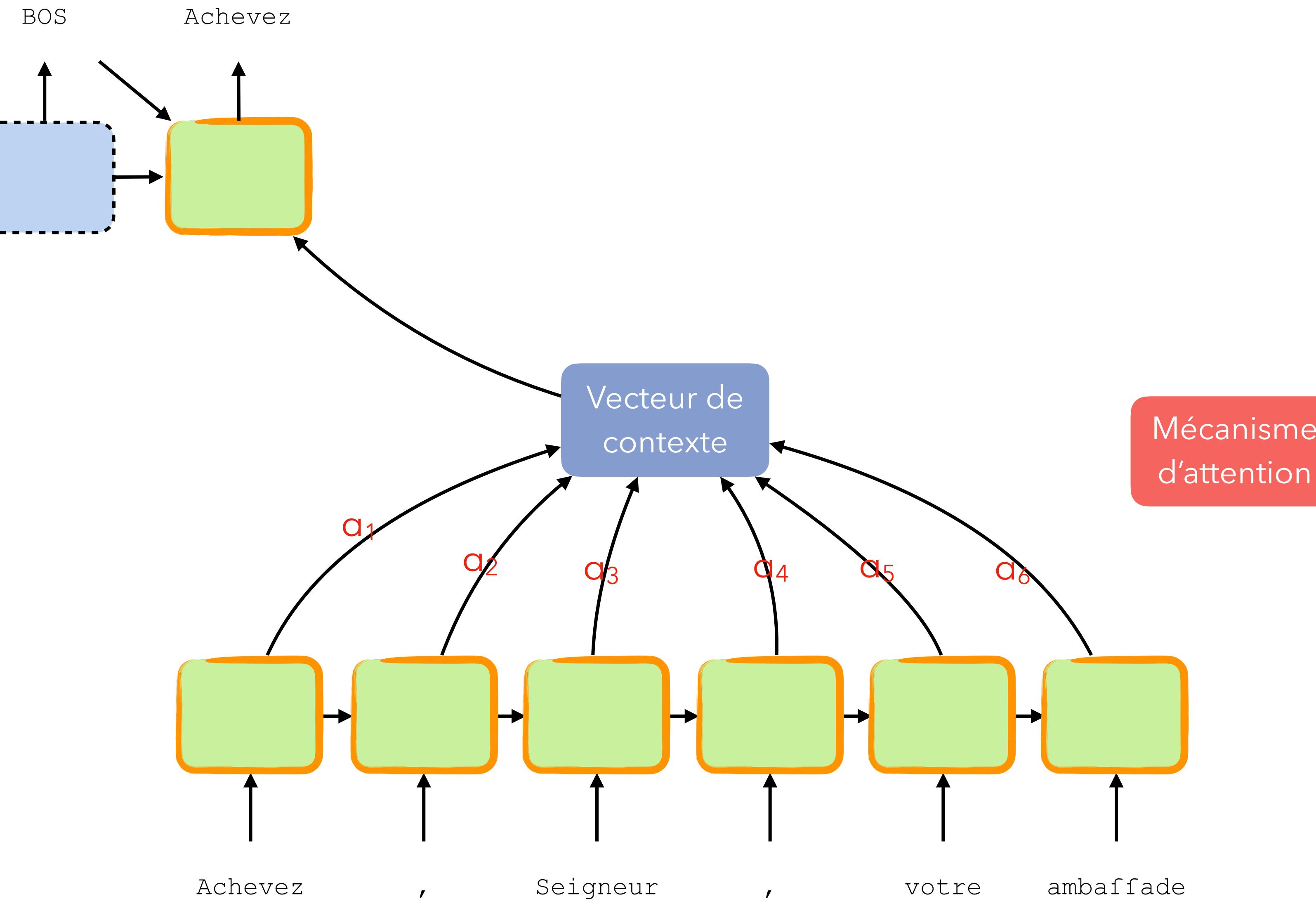
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



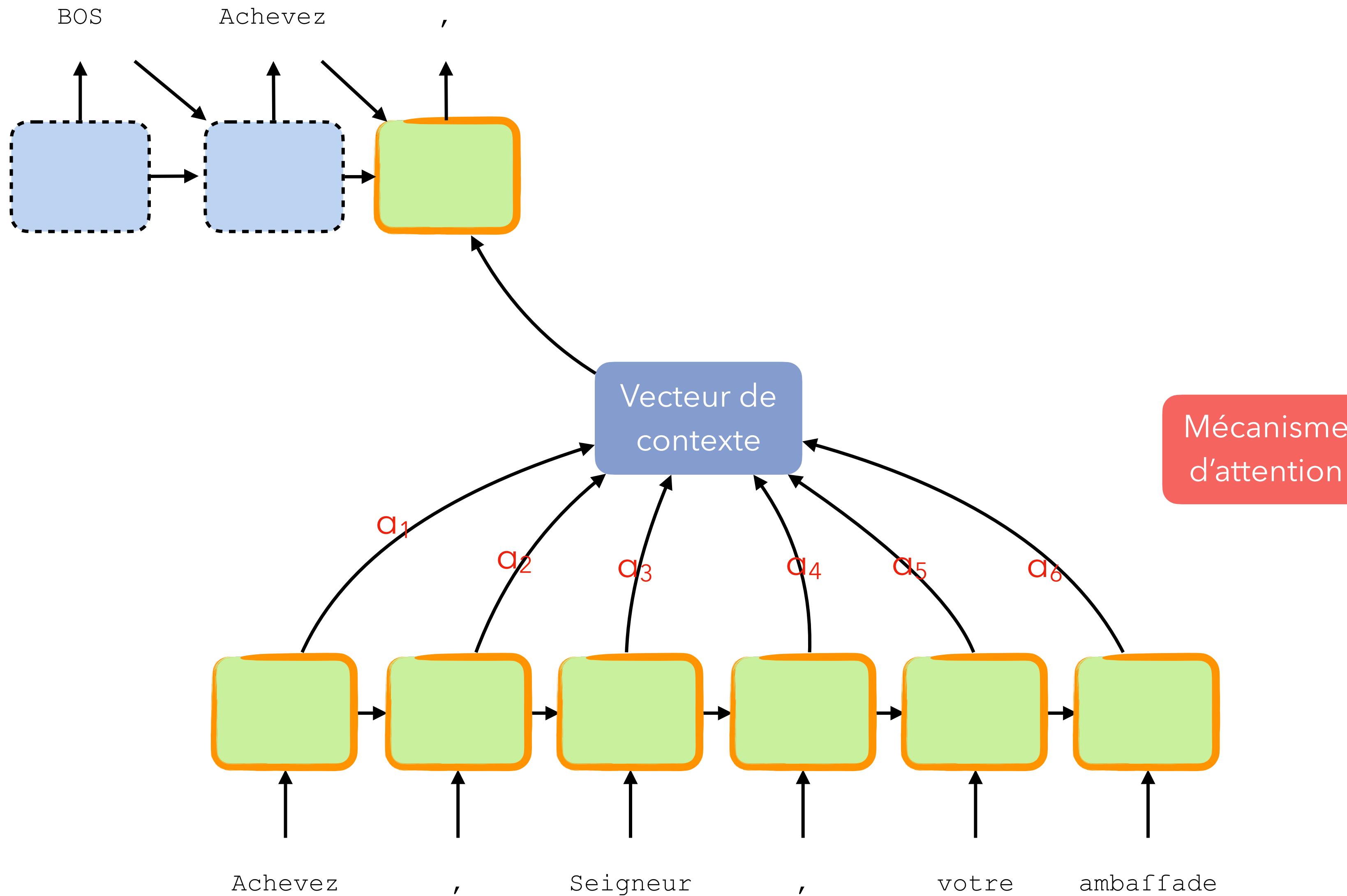
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



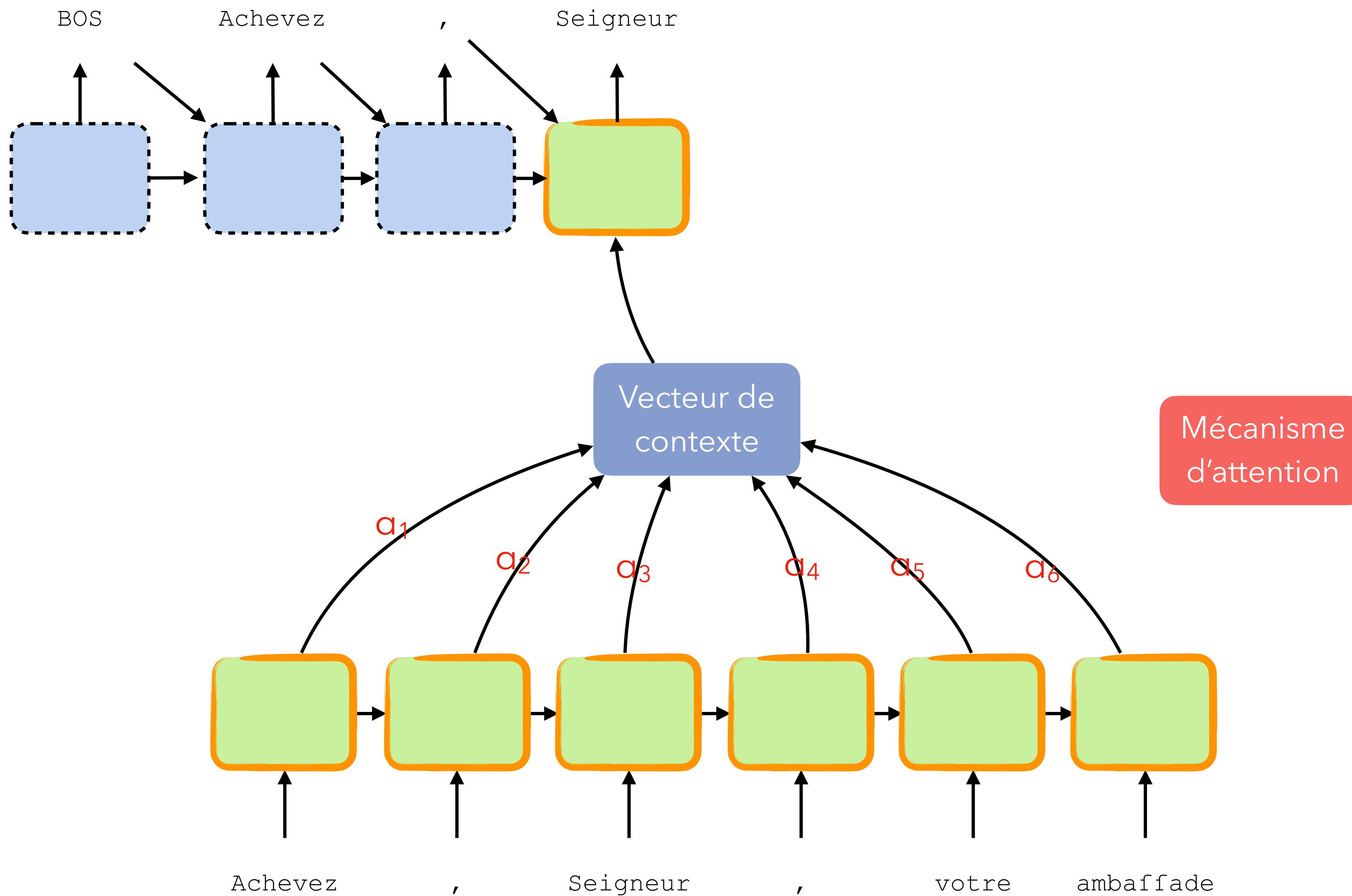
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



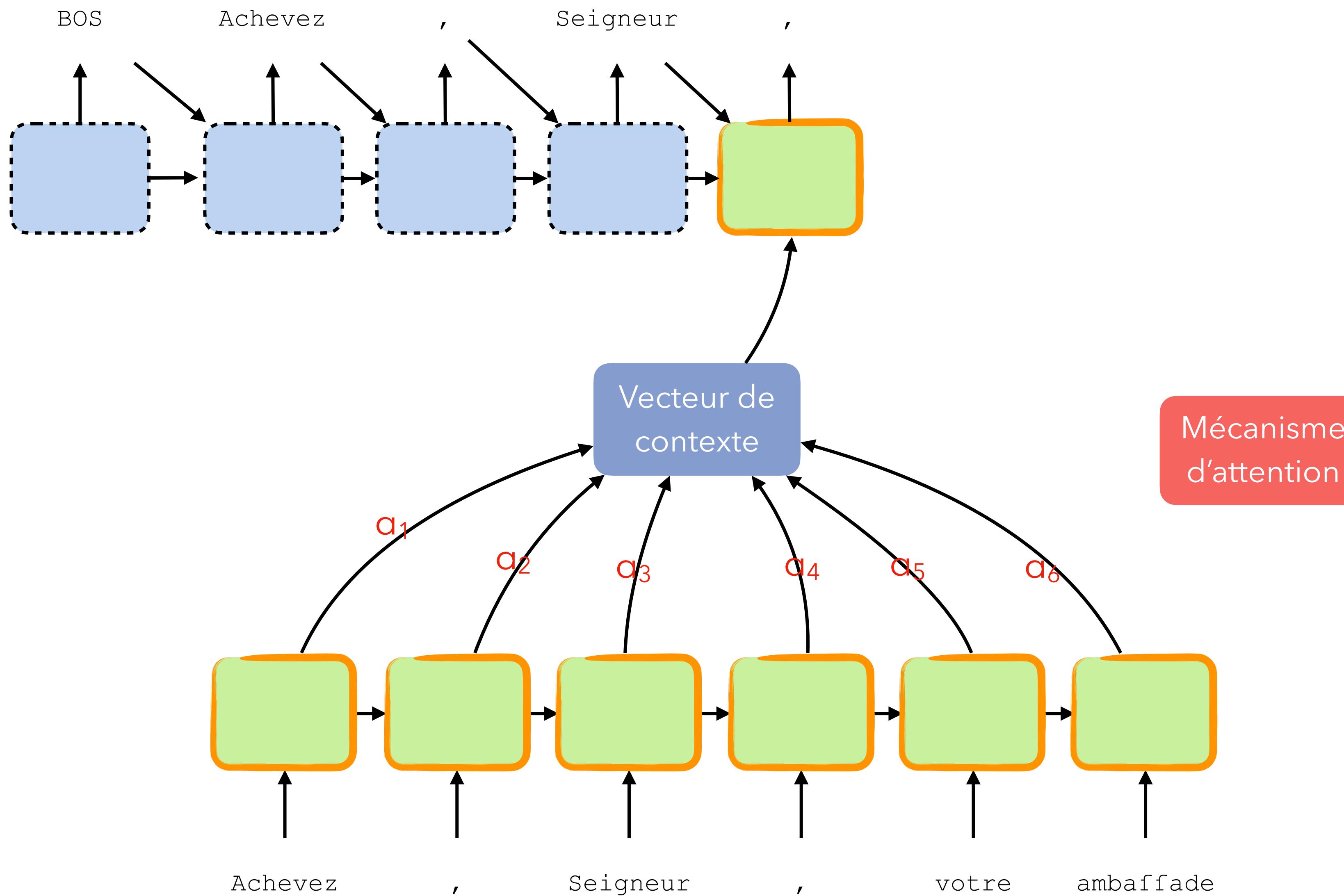
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



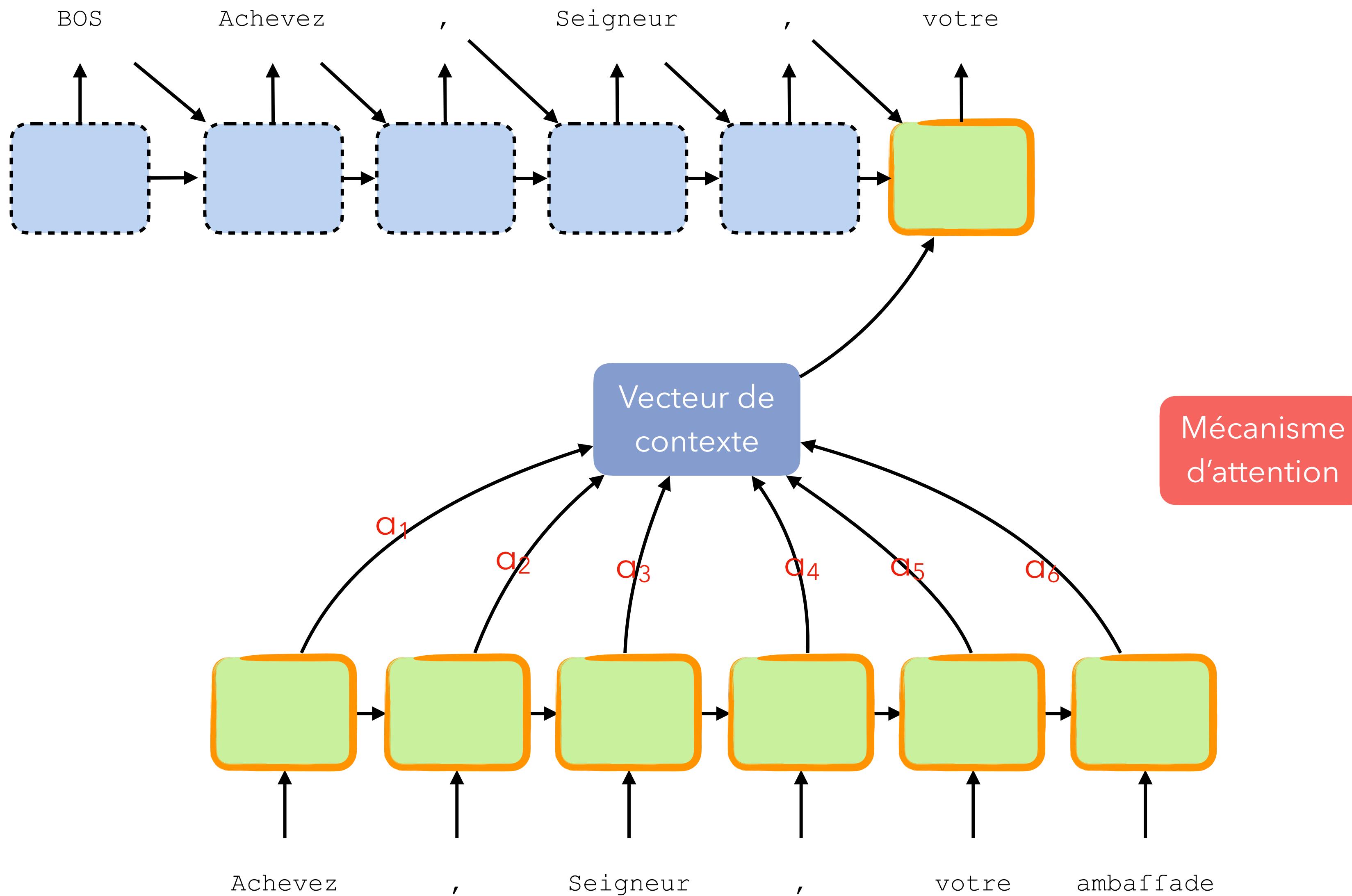
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



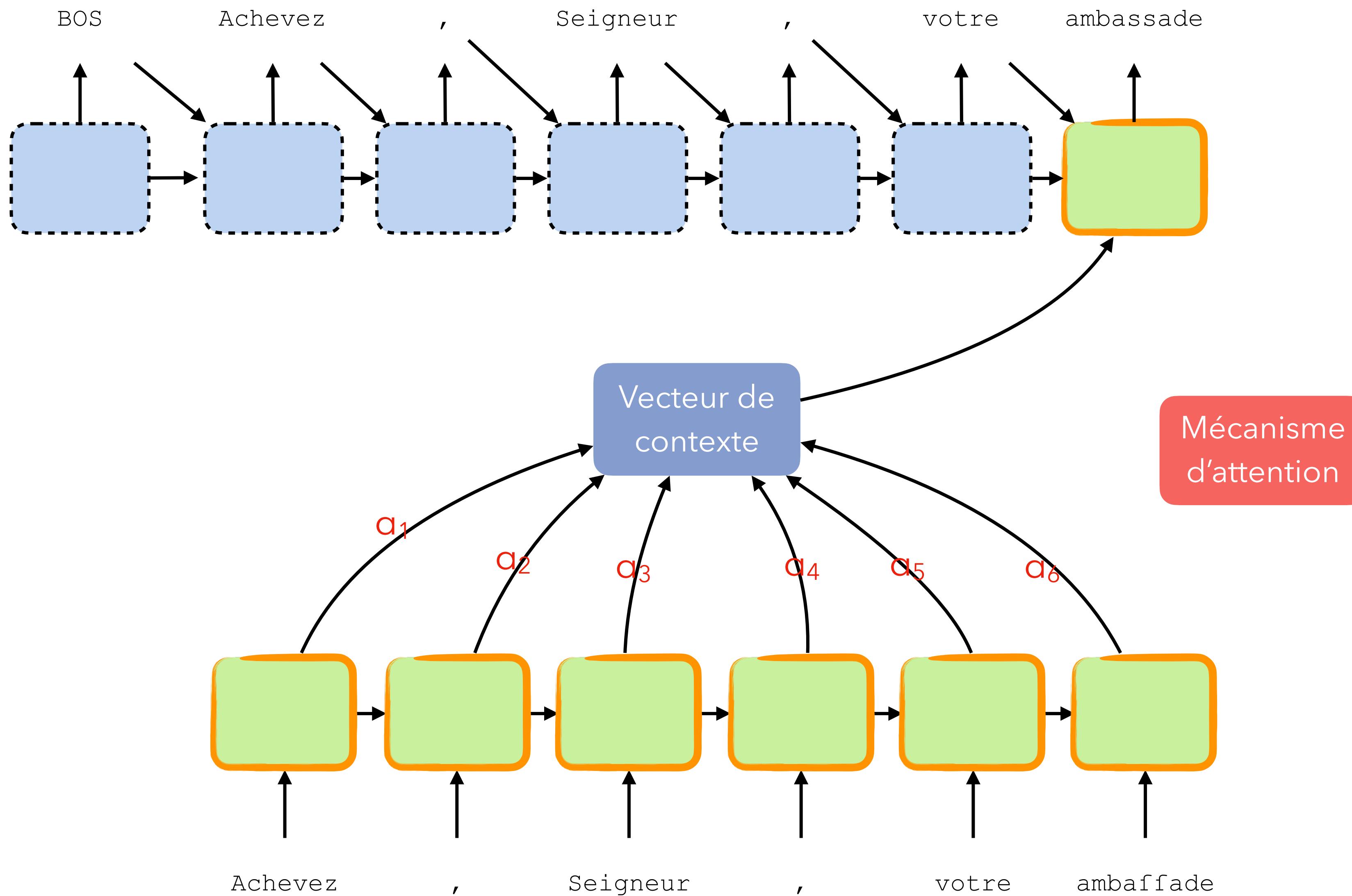
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



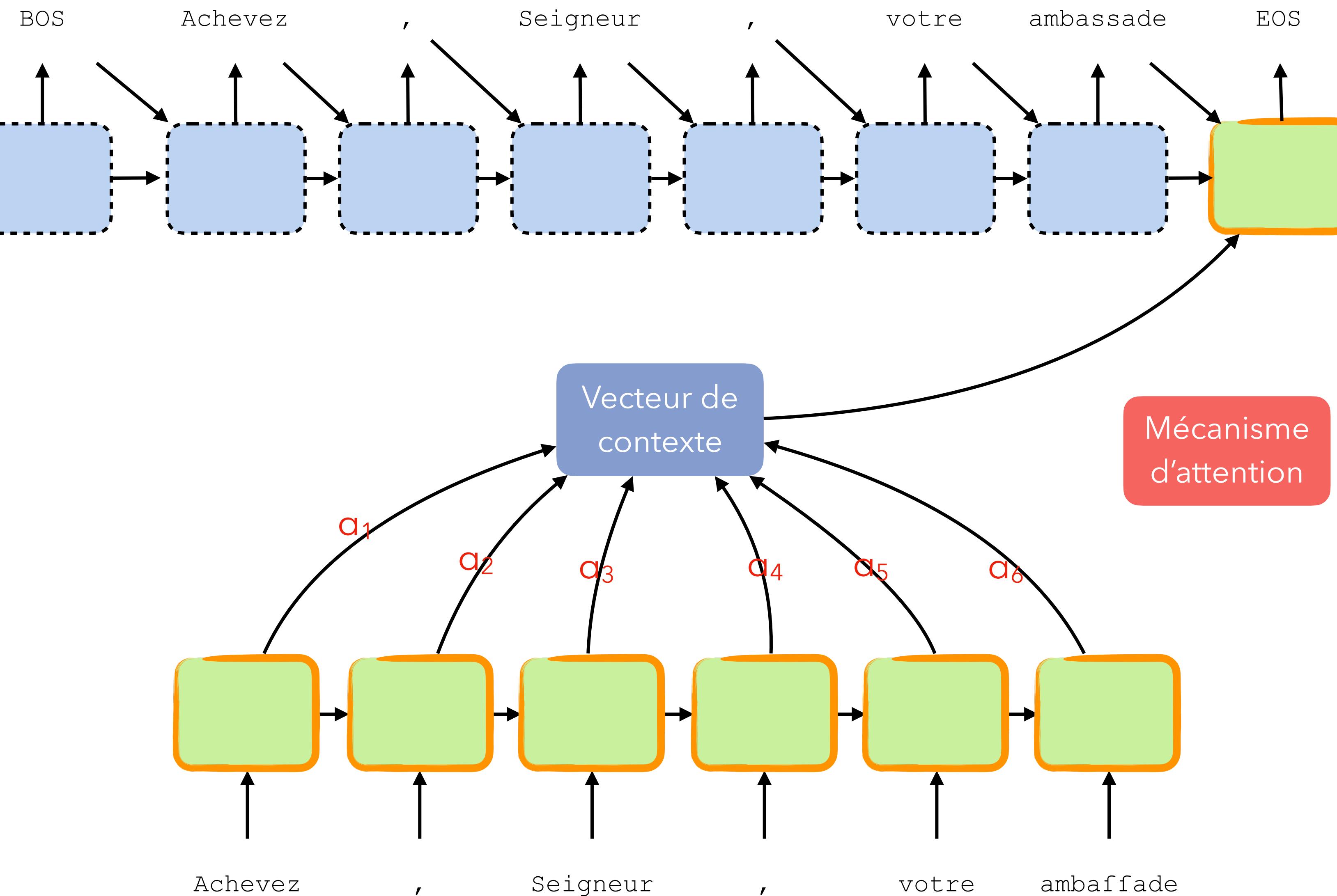
- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention



- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Amélioration : mécanisme d'attention

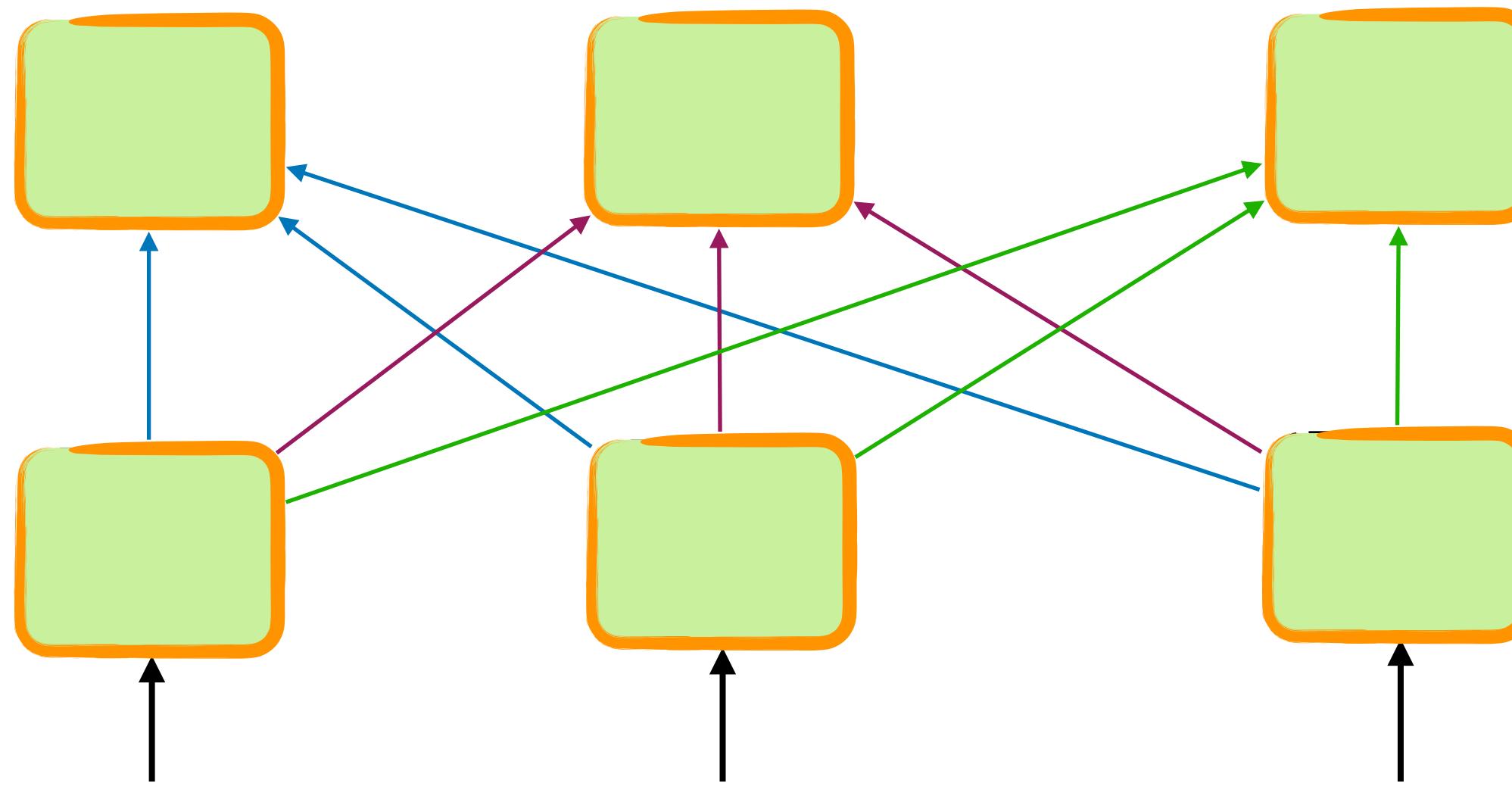


- Utiliser une représentation différente et adaptée du texte d'entrée pour chaque étape de décodage
- Les représentations d'entrées sont pondérées (certains mots sont plus importants pour produire certains mots)

# Le transformer : calcul en parallèle



- Un modèle encodeur-décodeur comme avant
- Mais remplace la récurrence par des calculs parallèles (+ quelques autres différences !)
  - Le self-attention : la représentation de chaque mot se fait à partir d'une somme pondérée des représentations de tous les mots du texte



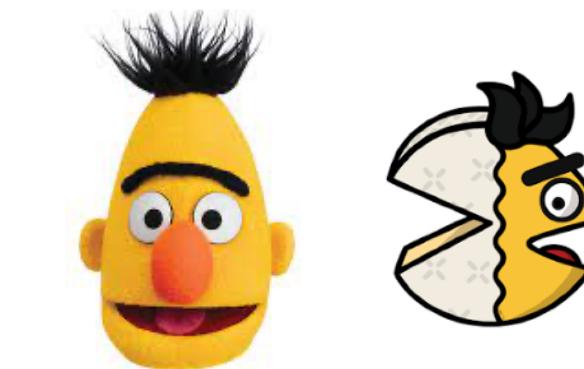
Achevez

,

Seigneur

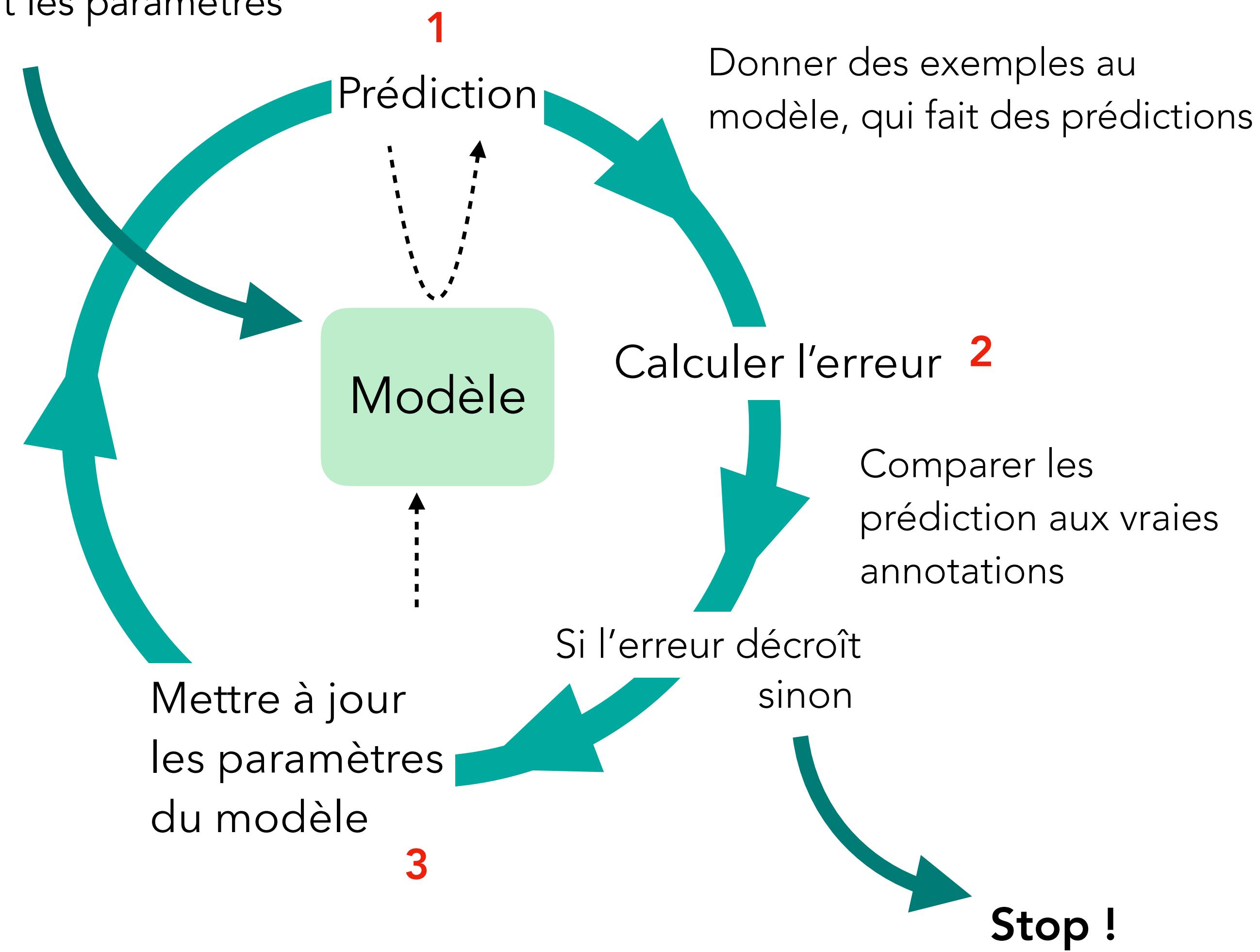
- Avantages :

- Le modèle peut s'entraîner plus vite !
- Des mots éloignés dans la phrase peuvent toujours dépendre les uns des autres



# Comment est-ce que le modèle apprend?

0 Initialiser aléatoirement les paramètres



Données divisées en 3 parties :

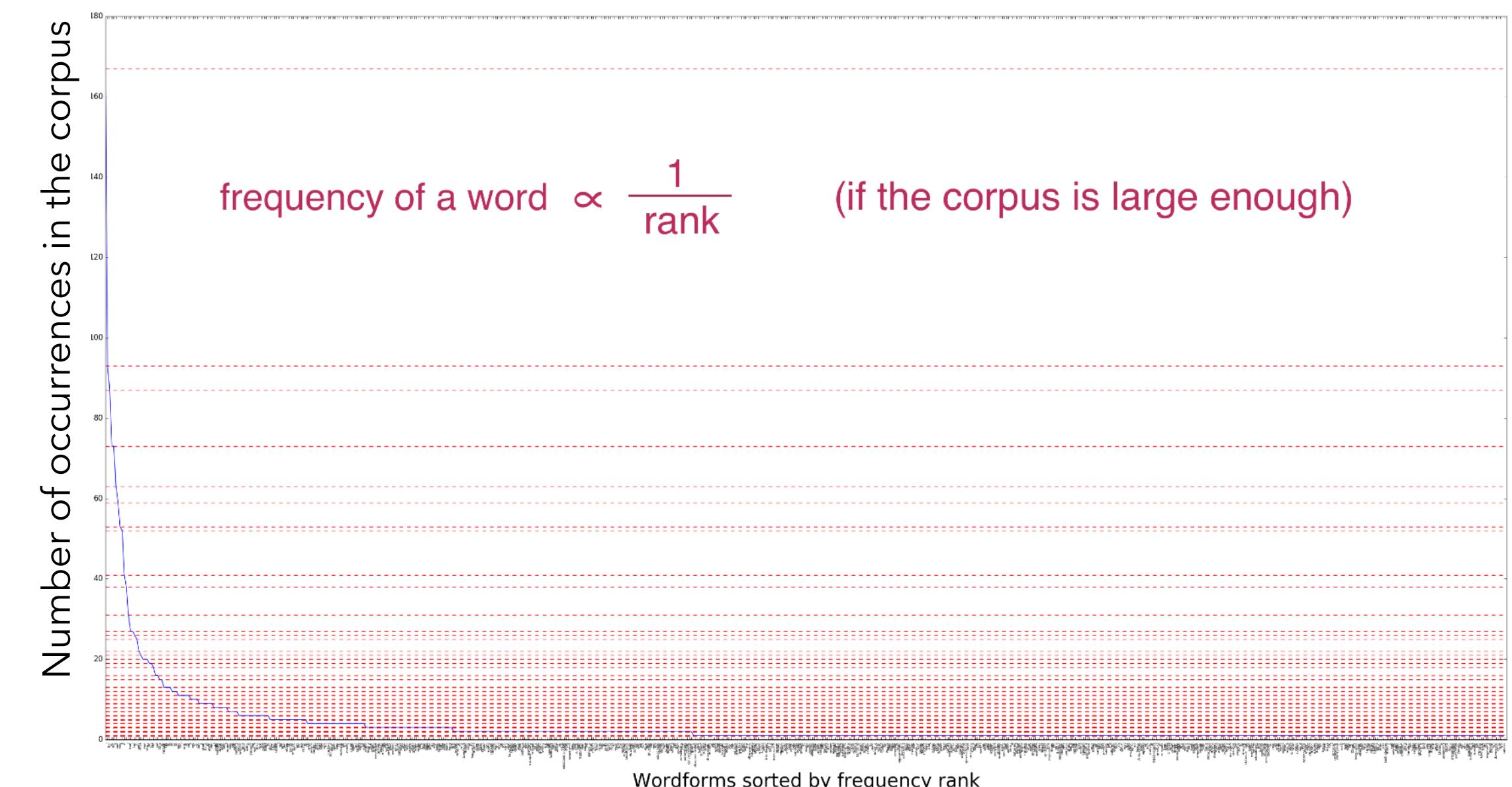
- Entraînement (train)
- Développement (dev)
- Test (test)

# Pré-traitement de textes

- Un modèle neuronal nécessite de connaître à l'avance les unités d'entrée, Le nombre d'unités (la taille du vocabulaire) est fixe
- Ceci est problématique pour traiter le texte, car:
  - Il y a beaucoup de mots (ça fait un vocabulaire et donc un modèle très gros)
  - Beaucoup de mots sont très rares et donc les embeddings appris par le modèle sont mal appris
  - On ne peut jamais couvrir tous les mots d'une langue (Loi de Zipf)

# Pré-traitement de textes

- Un modèle neuronal nécessite de connaître à l'avance les unités d'entrée, Le nombre d'unités (la taille du vocabulaire) est fixe
- Ceci est problématique pour traiter le texte, car:
  - Il y a beaucoup de mots (ça fait un vocabulaire et donc un modèle très gros)
  - Beaucoup de mots sont très rares et donc les embeddings appris par le modèle sont mal appris
  - On ne peut jamais couvrir tous les mots d'une langue (Loi de Zipf)



# Pré-traitement de textes

## Segmentation en sous-mots

- Une solution est de segmenter les mots en sous-mots
  - Ex : *the establishments were hungry* -> the establish ment s were hung r y
- Avantages linguistiques:
  - Permet la généralisation de « sens/fonction » sur de plus petites unités de texte, même sur les mots non précédemment vus
  - Dans beaucoup de cas, n'importe quel mot d'une langue peut être représenté par une combinaison de sous-mots
- Avantages computationnels : permet un vocabulaire plus petit

# Pré-traitement de textes

## Segmentation en sous-mots

- Une segmentation morphologiquement ou statistiquement motivée,
  - Une méthode populaire = BPE (*byte pair encoding*)
    - Traiter le texte d'entraînement comme une séquence de caractères individuels
    - Fusionner successivement la séquence de caractères la plus fréquente (un certain nombre de fois, défini à l'avance)
    - Résultat = les mots fréquents sont des unités en soi, les mots rares sont décomposés en sous-mots ou même en caractères
  - Nous allons voir tout à l'heure cet algorithme utilisé avec le toolkit sentencepiece  
Et dont , toutes les fois que je I ' ent end s parler .  
(les espaces sont traités comme des caractères et donc sont représentés par des \_)

# 3. Partie expérimentale

# Entraîner un modèle de normalisation

- Pour « traduire » le français moderne vers le français contemporain
- en utilisant des architectures de traduction standard
- ...et des données parallèles : des textes en français moderne où chaque phrase est associée à sa version normalisée

# Données

- PARALLEL17 :
  - Corpus parallèle : des textes du XVII<sup>ème</sup> et leurs normalisations en français contemporain
  - Des textes variés : prose, vers, différents genres (surtout littéraires) et distribués diachroniquement par décennie
  - Train : 17 930 phrases, dev : 2 443 phrases, test : 5 706
- D'AlemBERT
  - Très grand corpus de textes du XVII<sup>ème</sup> (mais pas parallèle)
  - Certains des textes ont leur graphie d'origine, certains sont écrits selon la norme du français contemporain

# Comparaison de modèles

- Comparaison entre modèles récurrents et transformers
  - Les transformers sont état de l'art, mais fonctionne parfois moins bien quand il y a moins de données d'entraînement
- Comparaison de différents tailles de vocabulaires (à quel point on segmente le texte en sous-mots)
  - Une segmentation plus fine marche souvent mieux lorsqu'il y a peu de données d'entraînement, mais une trop forte segmentation mène à des séquences très longues
- Comparaison entre plusieurs tailles d'architectures
  - On peut changer le nombre de couches, la taille des représentations, etc.
  - Utiliser plus de paramètres améliore la capacité du modèle à bien modéliser, mais ça nécessite souvent plus de données !

# Évaluation

- Il est important de savoir évaluer les modèles (automatiquement)
- Métriques courantes en traduction automatique :
  - BLEU : s'appuie sur la présence des mêmes séquences de mots ( $n$ -grams) dans un texte prédit par rapport à son texte « de référence »
  - ChrF : similaire au BLEU, sauf qu'il se base sur les  $n$ -grams de caractères dans un texte prédit par rapport à son texte « de référence »
- Autres métrique (plus adaptés) :
  - Exactitude au niveau de mots individuels

# Exactitude au niveau des mots

- Nous ne nous attendons pas à devoir changer l'ordre des mots
- Donc il suffit d'**aligner les mots du texte cible** (de référence) **et les mots du texte prédit** et comparer mot à mot pour calculer le nombre de bonnes prédictions (y compris les mots qui n'étaient pas à changer)
- Utilisation de la distance de Levenshtein, qui produit un alignment entre deux séquences de texte

# Levenshtein

	f	r	e	r	e		e	f	t	o	i	t
f												
r												
è												
r												
e												
é												
t												
a												
i												
t												



Même caractère

## Éditions: ce qui « coûte » !



Substitution de caractère

Insertion de caractère

Suppression de caractère

- Chercher le « chemin » dans la matrice d'édition qui minimise le nombre d'éditions nécessaire
- Fournit une distance entre les deux séquences (ici = 4)
- Mais fournit aussi un alignement entre les deux textes

# Exactitude à base de l'alignement produit

	f	r	e	r	e		e	f	t	o	i	t
f												
r												
è												
r												
e												
é												
t												
a												
i												
t												

- Calculer l'alignement entre la phrase prédictée et la phrase de référence (au niveau des caractères)
- Découper le phrase de référence en mots à partir de l'alignement en caractères, trouver les mots de la prédiction qui correspondent :  
frere → frère  
était → eftoit
- Calculer combien de mots sont les mêmes, divisé par le nombre total de mots = exactitude (accuracy)

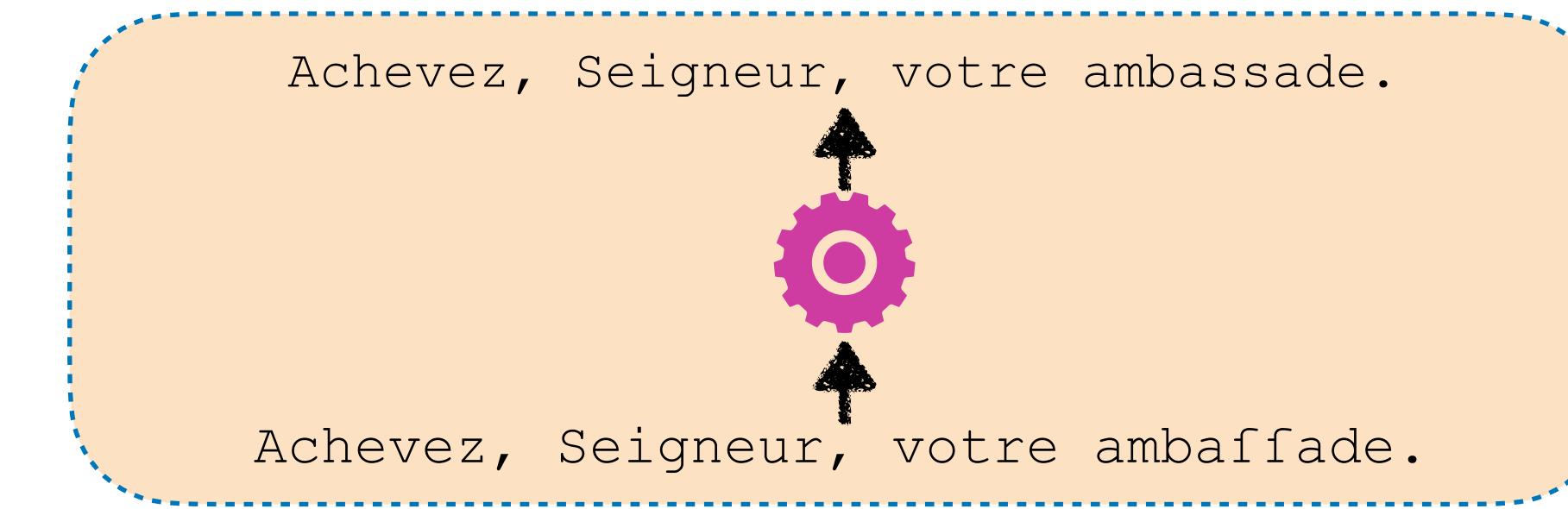
# Résultats

Modèle	Exactitude	BLEU	ChrF
Identité (ne rien changer)	73,6 %	42,33	75,0 %
Approche à base de règles	89,5 %	74,26	90,5 %
Approche à base de règles + lexique	91,3 %	78,91	92,4 %
Méthode ABA (en se basant sur les alignements automatiques)	95,3 %	89,2	96,4 %
LSTM	96,9 %	92,98	97,5 %
Transformer	96,4 %	91,90	97,1 %

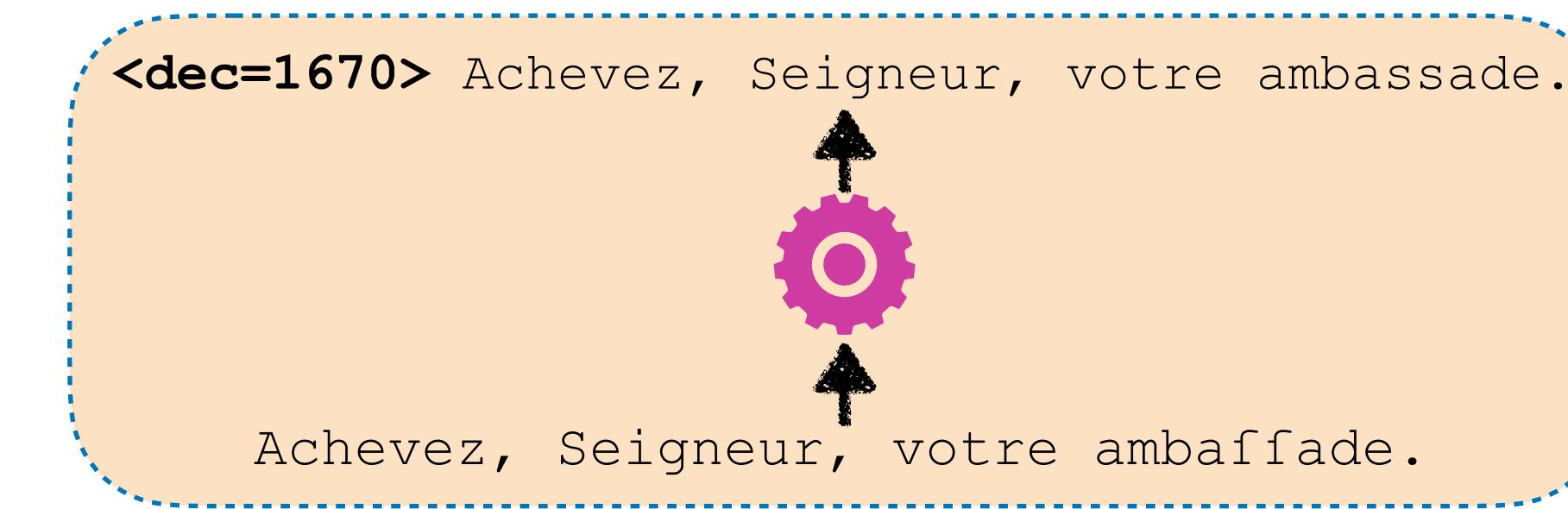
- Les baselines arrivent déjà à avoir de « bons scores »
- Le modèle LSTM (récurrent) marche mieux que le transformer)

Au-delà de la normalisation :  
Suivre le développement linguistique/graphique au cours du siècle

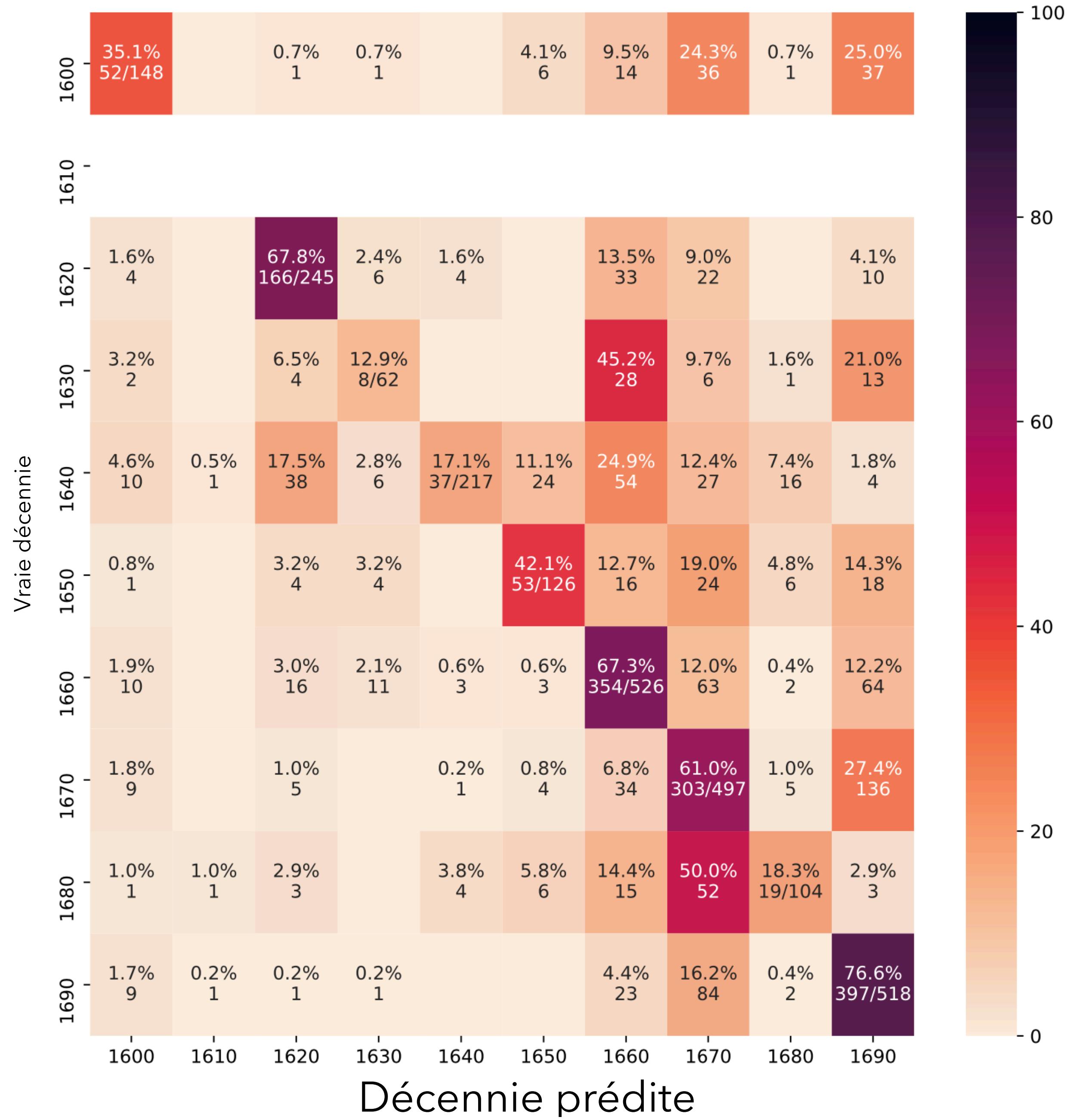
# 1. Sensibilité au changement linguistique/graphique



Y a-t-il suffisamment d'informations dans les phrases originales pour prédire le texte normalisé et la décennie de rédaction ?



Apprendre au modèle de prédire la décennie de rédaction du texte à partir de la phrase d'entrée. La décennie de rédaction est traité comme un mot spécial au début de la phrase de sortie



# Résultats

- Comparaison entre les décennies prédictes (en bas) et la vraie décennie (à gauche)
- En général, de bonnes prédictions, même si les décennies 1660 et 1670 sont sur-prédites

# 1. Sensibilité au changement linguistique/graphique

Mais les informations apprises sont-elles de nature graphique ou lexicale ? (les textes ont du contenu différent, et de quoi parlent les textes pourraient être suffisant pour prédire la décennie de rédaction.

## Expérience de contrôle : dénormalisation (réduction de l'information au simple lexique)

The diagram consists of two text snippets within an orange rounded rectangle, connected by two black arrows pointing upwards towards a central purple gear icon. The top snippet contains the text '<dec=1670> Achevez, Seigneur, votre ambaffade.' and the bottom snippet contains the text 'Achevez, Seigneur, votre ambassade.' A blue dashed oval surrounds the entire diagram.

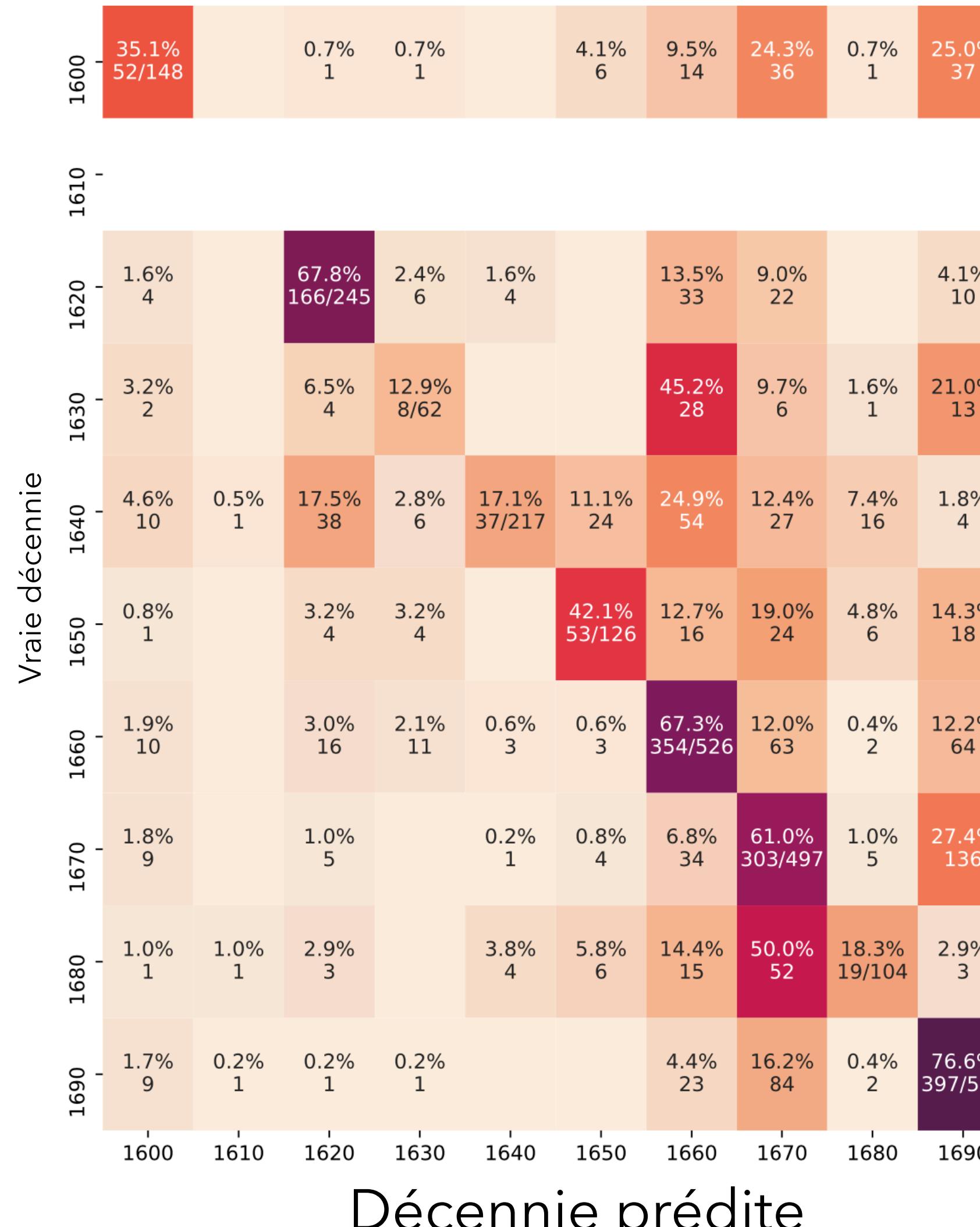
<dec=1670> Achevez, Seigneur, votre ambaffade.

Achevez, Seigneur, votre ambassade.

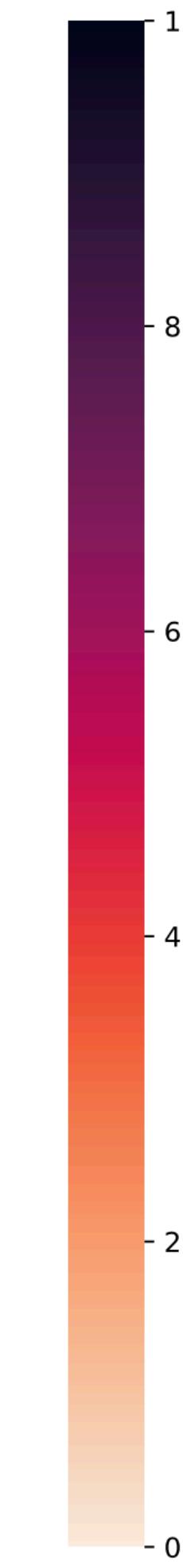
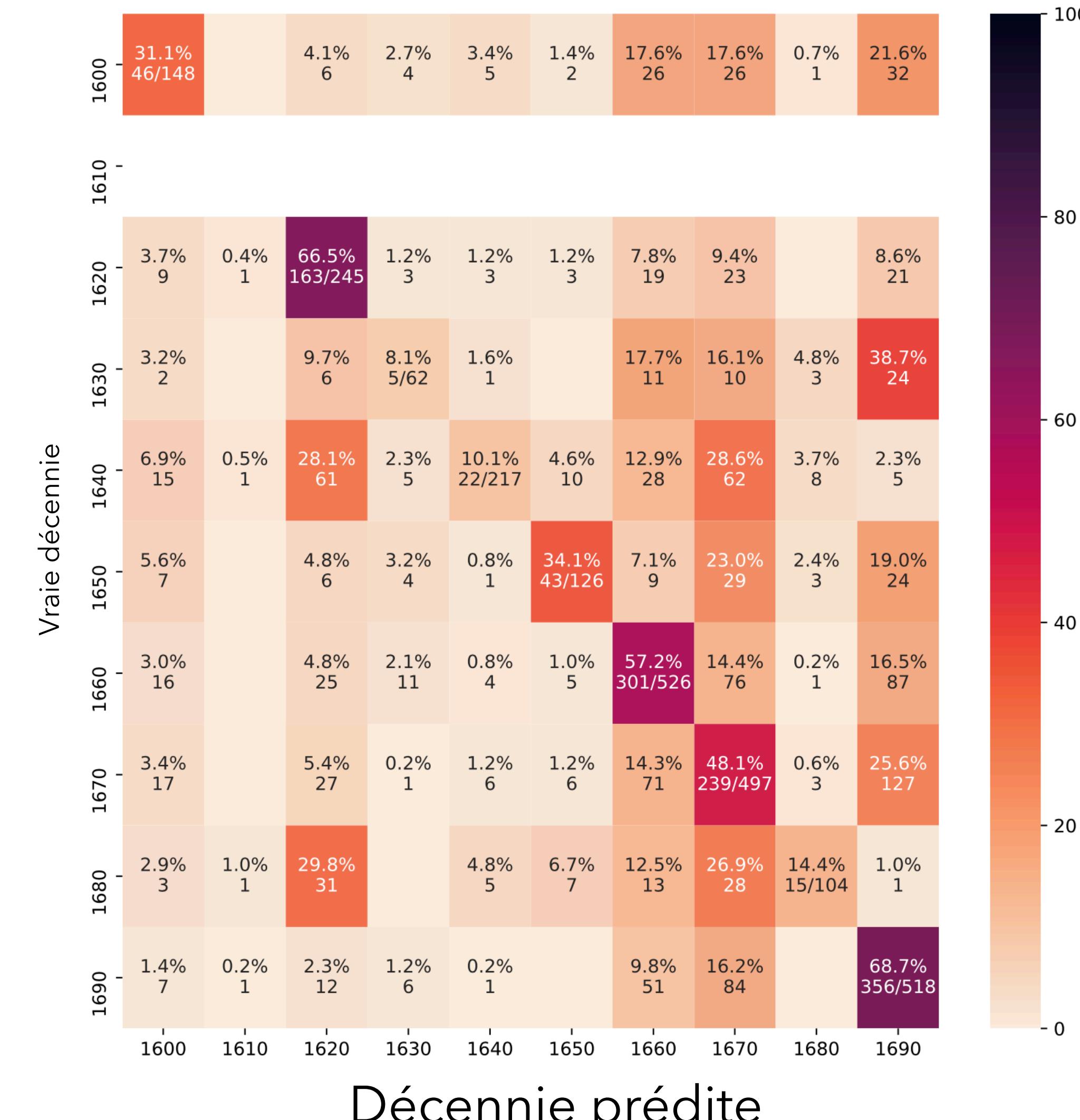
- Apprendre à prédire le français moderne à partir de la version normalisée du texte (en français contemporain)
- Comme avant, prédire aussi la décennie

# Résultats : comparaison

Normalisation : 57%



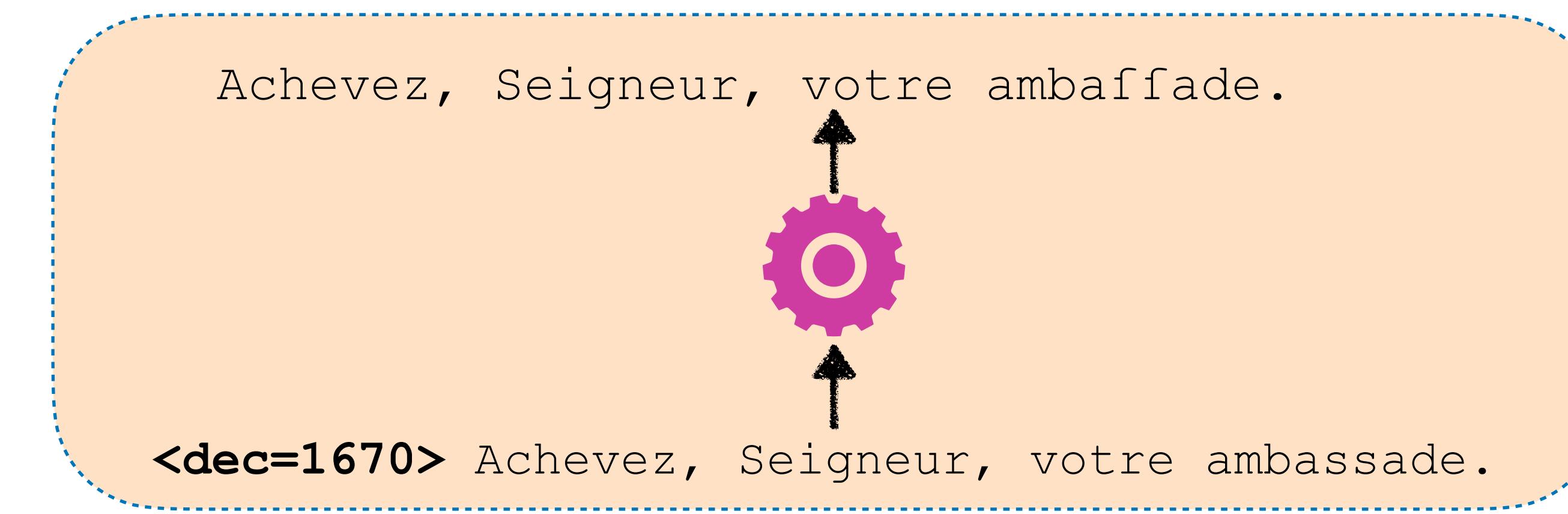
Dénormalisation : 49%



- Prédictions bonnes pour les 2 modèles
- Les scores sont plus élevés quand le modèle a accès à la variation graphique
- Le modèle ne s'appuie donc pas uniquement sur le lexique pour prédire la décennie

# Quelles informations sont apprises ?

- On aimerait extraire les connaissances apprises par le modèle pour les comparer à différentes périodes du siècle
- Entraîner un modèle de dénormalisation, en conditionnant la prédiction sur la décennie de rédaction — cela permet de contrôler quel type de texte est produit



# Création de multiples corpus comparables

- Modèle de dénormalisation, conditionné sur la décennie
  - Nous pouvons donc créer une version « classique » artificielle/ synthétique pour chaque décennie (1600, 1610, 1620...)
  - L'objectif est de concevoir un corpus parfaitement homogène, où chaque décennie est parfaitement comparable à une autre
- Nous pouvons faire appliquer ce modèle sur un grand nombre de données :
  - Nous prenons les textes déjà normalisés de D'AlemBERT

# Création de multiples corpus comparables

Qui fit **naître un** si bel **ouvrage** ?  
Ta Muse a triomphé du temps ;  
Son couchant est **une** autre aurore ;  
**Grave**, ou **folâtre**, elle est encore  
Ce qu' elle **était** dans son printemps

(43k sentences)

→ 1670-1679

...

...

...

→ 1690-1699

Qui fit **naistre vn** si bel **ouurage** ?  
Ta Muse a triomphé du temps ;  
Son couchant est **vne** autre aurore ;  
**Graue**, ou **folastre**, elle est encore  
Ce qu' elle **estoit** dans son printemps

Qui fit **naistre un** si bel **ouvrage** ?  
Ta Muse a triomphé du temps ;  
Son couchant est **une** autre aurore ;  
**Grave**, ou **folastre**, elle est encore  
Ce qu' elle **estoit** dans son printemps

Qui fit **naître un** si bel **ouvrage** ?  
Ta Muse a triomphé du temps ;  
Son couchant est **une** autre aurore ;  
**Grave**, ou **folastre**, elle est encore  
Ce qu' elle **estoit** dans son printemps

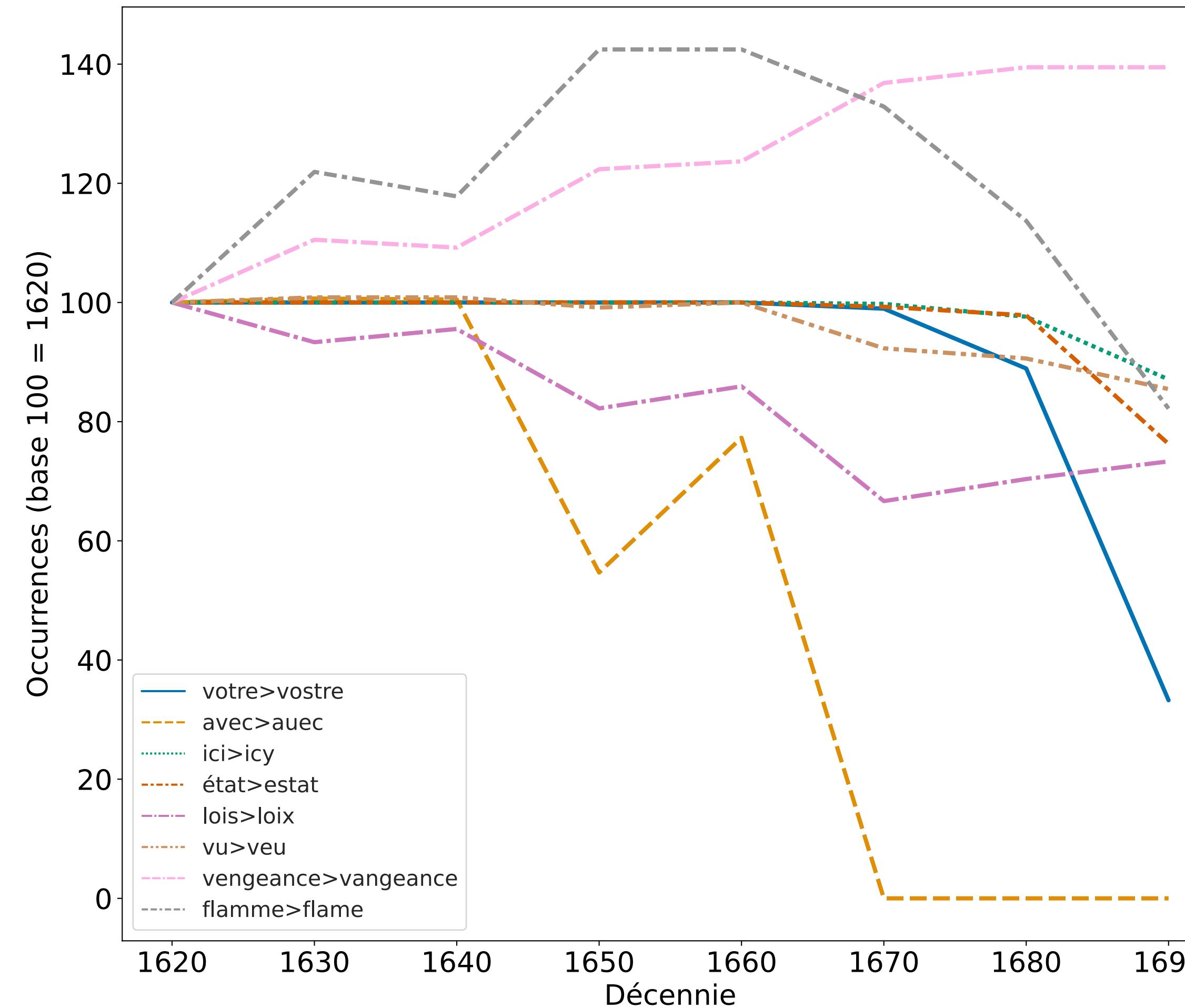
naître → naistre	1
un → vn	1
ouvrage → ouurage	1
une → vne	1
Grave → Graue	1
folâtre → folastre	1
était → estoit	1

naître → naistre	1
folâtre → folastre	1
était → estoit	1

folâtre → folastre	1
était → estoit	1

Comparaison  
du nombre  
d'occurrences

# Analyser les correspondances



- Compter le nombre de chaque changement
  - Visualiser quelles correspondances varient le plus au cours du siècle
- (variations en prenant comme base 100=1620)

## Ce qu'on voit :

- Les changements dans les conventions d'écriture (orthographies classiques vs. modernes)
- Aussi du changement linguistique
- Des preuves pour des points de changement dans le temps (ex. : 1670)

# Conclusion

- Déjà de bons résultats en normalisation en utilisant les modèles de traduction automatique → mais il reste des erreurs !
- Nous pouvons aussi exploiter ces modèles pour analyser des textes, en essayant d'extraire les connaissances apprises
- Nous comptons aller plus loin :
  - à la recherche d'une signature d'un texte
  - exploiter les autres méta-information (auteur, genre du texte, éditeur)

Merci ! Questions ?