

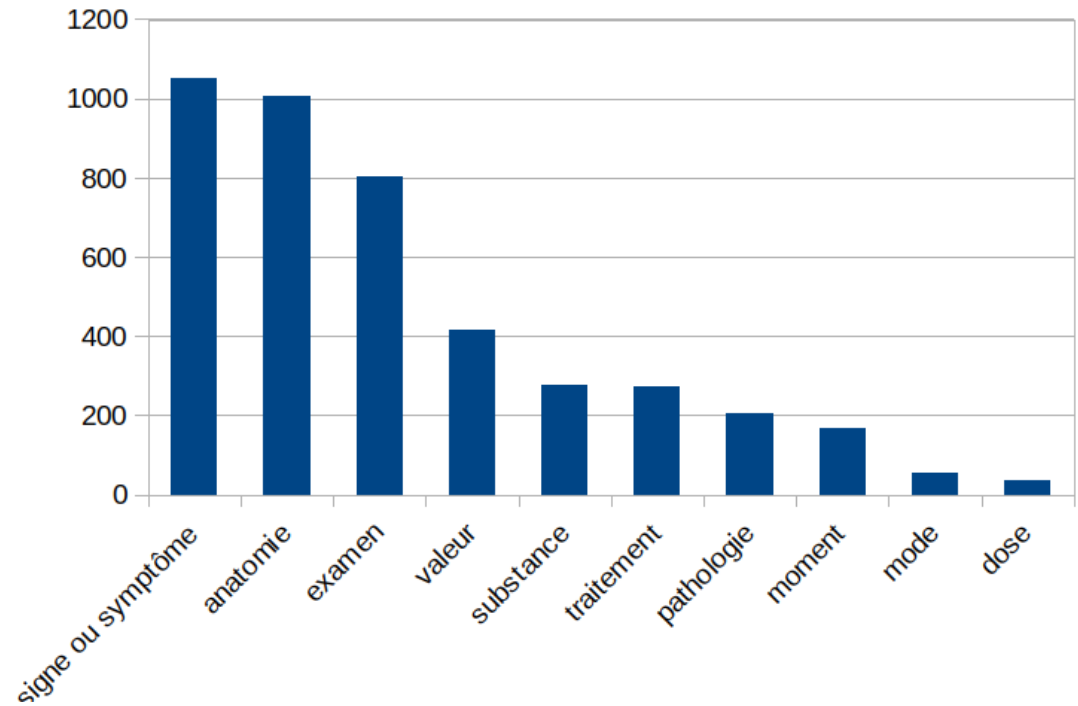
Participation à DEFT 2020

(tâche 3)

Perceval Wajsbürt, Yoann Taillé,
Guillaume Lainé, Xavier Tannier

Tâche 3

- <https://deft.limsi.fr/2020/>
- Une tâche de reconnaissance d'entités nommées :
 - Domaine de spécialité (cas cliniques)
 - Mentions imbriquées
- Jeu d'entraînement :
 - 67 documents
 - 3800 annotations



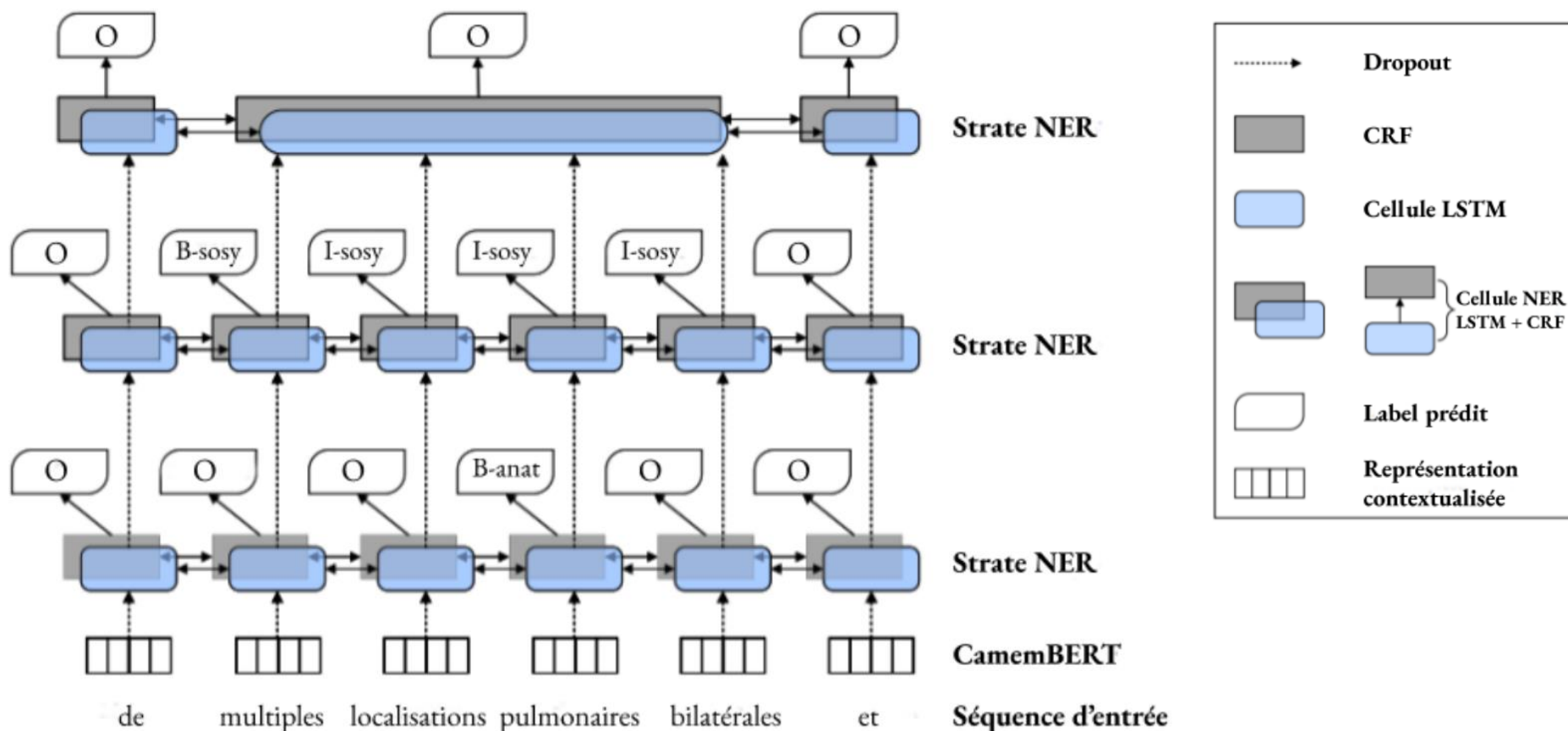
Deux approches

- Nous avons focalisé nos efforts sur la gestion des mentions imbriquées
- Premier système : plusieurs strates de modèles BiLSTM+CRF, une par « niveau d'imbrication ».
Fortement inspiré de *Ju et al. 2018*, A neural layered model for nested named entity recognition (NAACL)
- Second système : modèle itératif sans contrainte sur l'ordre des imbrications

Modèles mis en œuvre

1. Modèles BiLSTM+CRF en strates

- Fortement inspiré de *Ju et al. 2018*, A neural layered model for nested named entity recognition (NAACL)

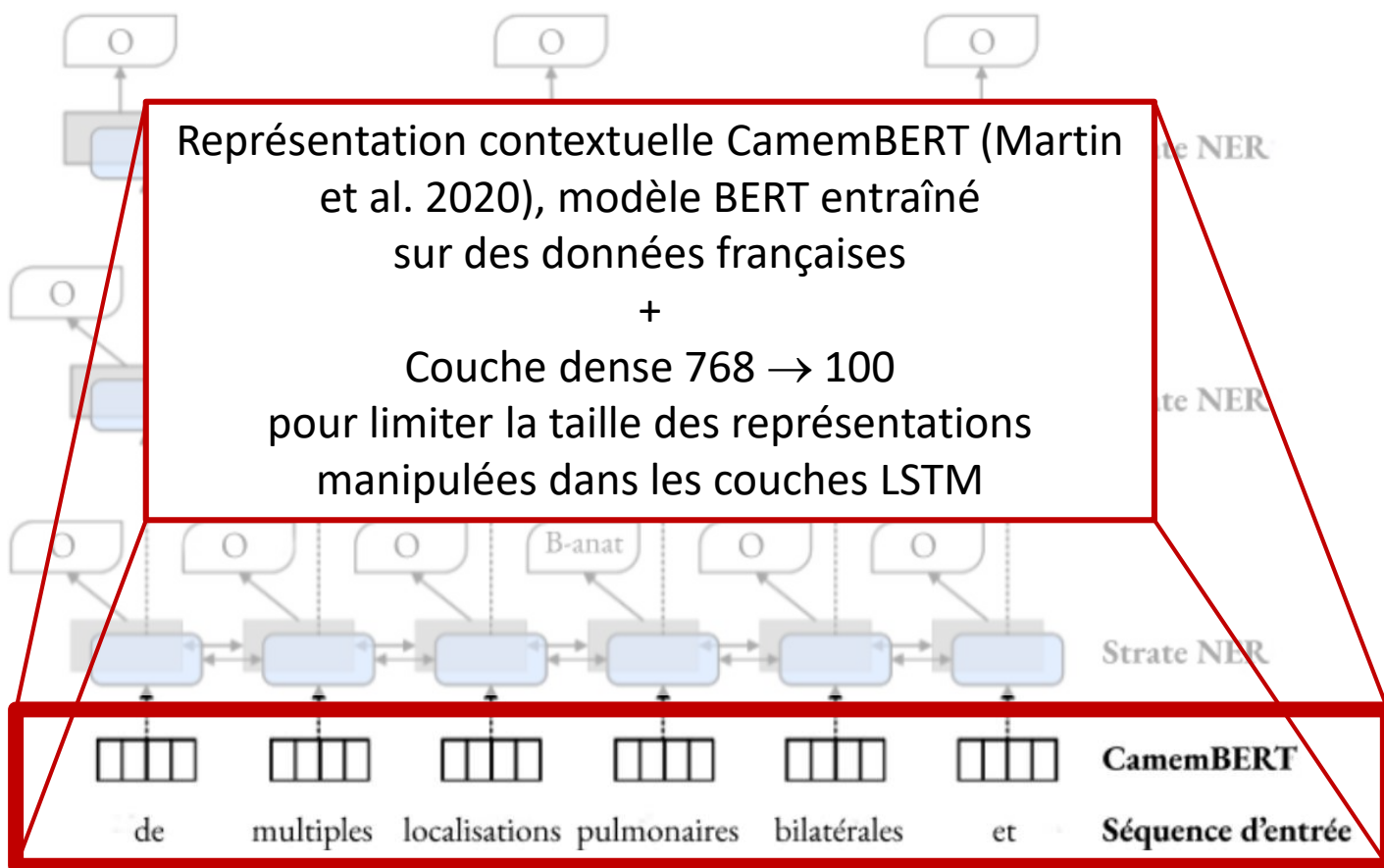
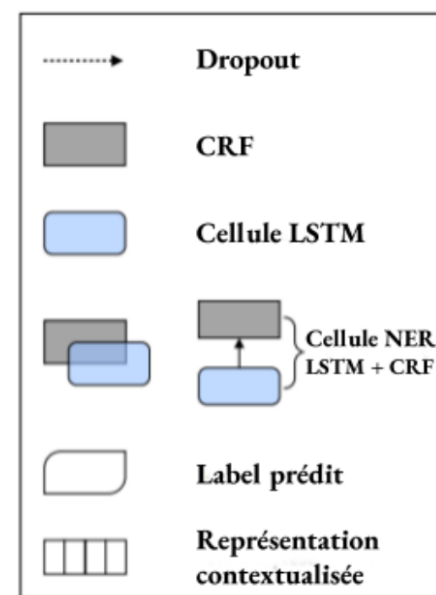


1. Modèles BiLSTM+CRF en strates

- Détection des mentions les plus courtes, puis des mentions plus longues les incluant
- Utilisation de la représentation classique BIO attribuant une classe à chaque token
 - B-ent : le token débute une mention de l'entité « ent »
 - I-ent : le token poursuit une mention de l'entité « ent »
 - O : le token n'est pas dans une mention d'entité
- Cette représentation ne supporte pas l'imbrication des mentions, d'où la superposition de plusieurs strates de reconnaissance.

1. Modèles BiLSTM+CRF en strates

Représentation contextuelle CamemBERT (Martin et al. 2020), modèle BERT entraîné sur des données françaises
+
Couche dense 768 \rightarrow 100
pour limiter la taille des représentations manipulées dans les couches LSTM

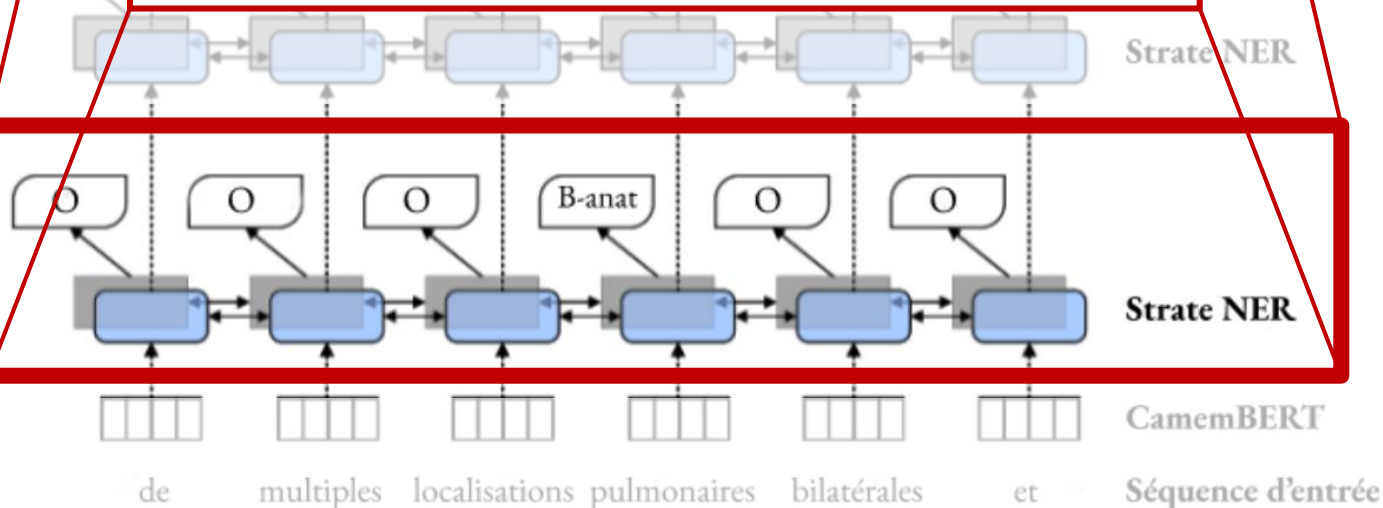


1. Modèles BiLSTM+CRF en strates

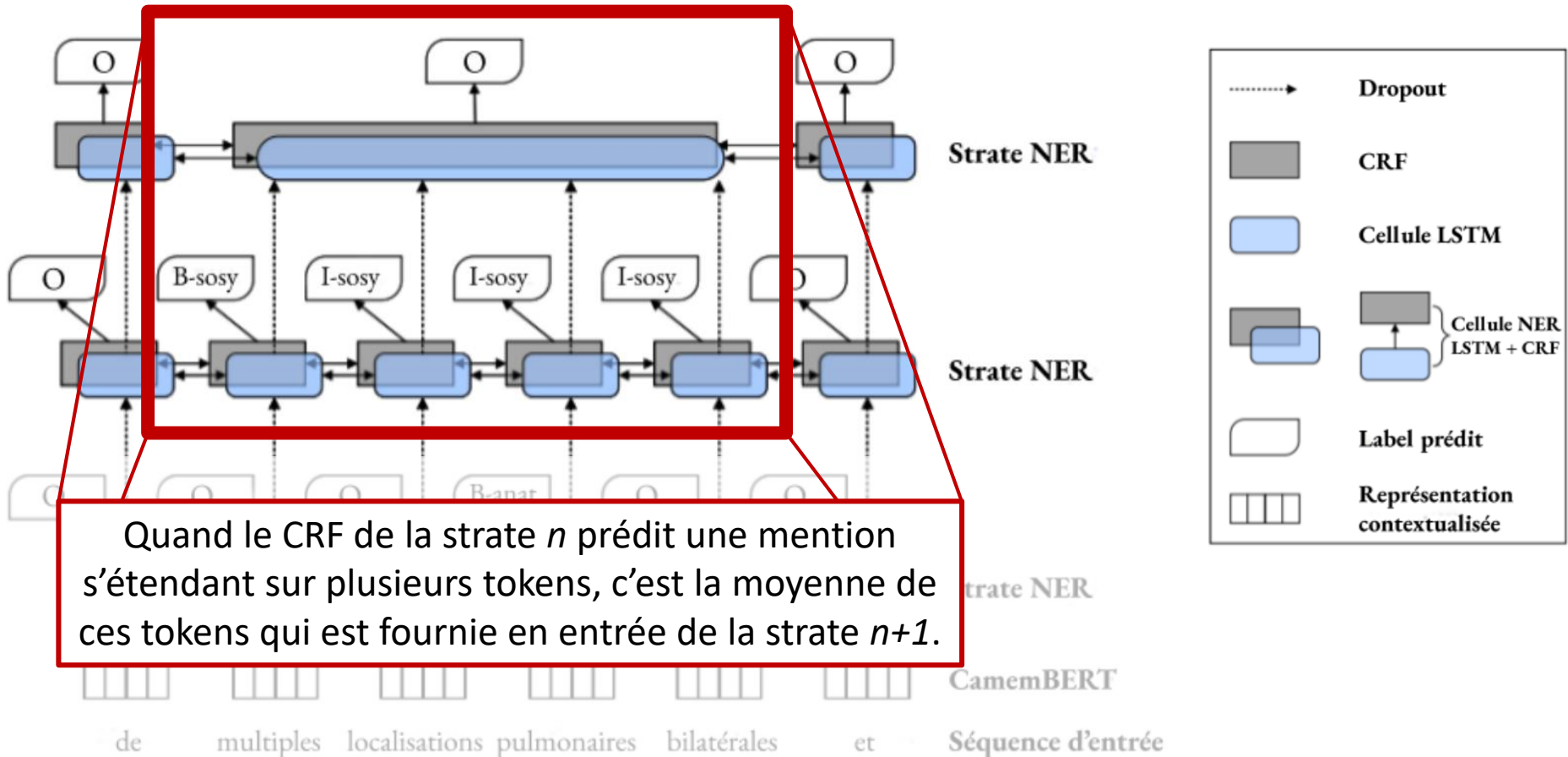
Chaque strate NER est une combinaison
LSTM bidirectionnel + CRF (*Lample et al., 2016*)

Entrée : représentation de la couche précédente
(embeddings ou strate du dessous)

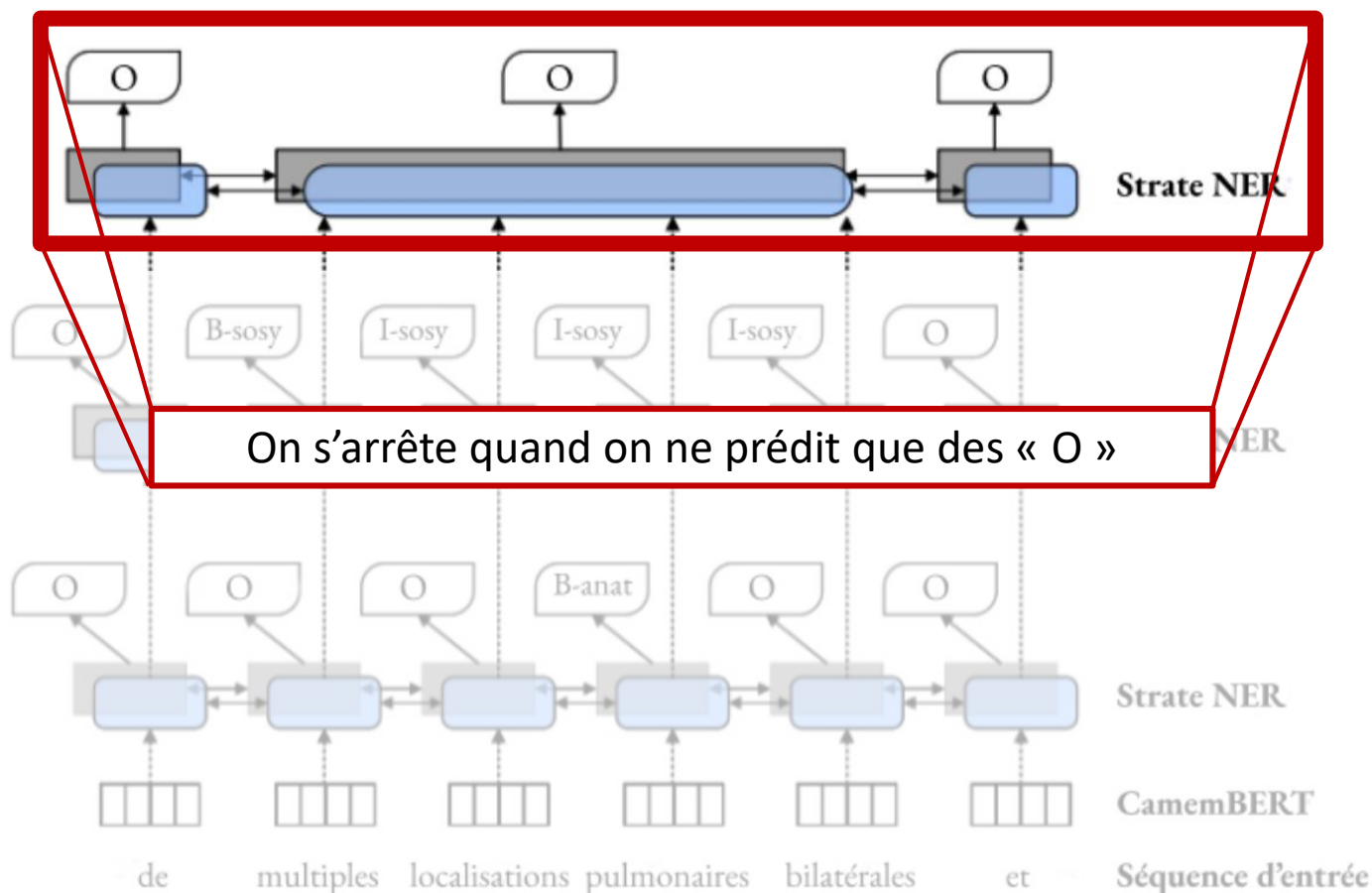
Sorties : une prédiction de mentions non imbriquées
+ une nouvelle représentation de chaque token



1. Modèles BiLSTM+CRF en strates



1. Modèles BiLSTM+CRF en strates



1. Modèles BiLSTM+CRF en strates

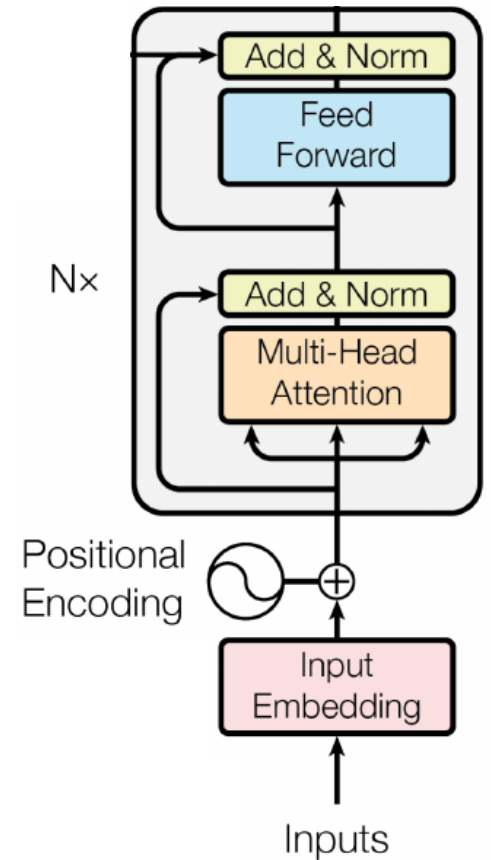
- Cette approche est adaptée à une tâche multi-classe, pourtant nos expériences préliminaires ont montré qu'entraîner 10 modèles binaires (un par type d'entité) plutôt qu'un seul modèle multi-classe était plus performant.
- Dans cette configuration, il n'y a jamais plus de 2 strates consécutives.
- Les hyper-paramètres et la méthode d'entraînement sont décrites dans l'article.

1. Modèles BiLSTM+CRF en strates

- Avantages de l'approche :
 - Simple à comprendre et à mettre en œuvre
 - N'importe quel modèle NER classique peut être branché
- Inconvénients :
 - Mentions totalement imbriquées uniquement (pas de chevauchement possible)
 - Unidirectionnel : les entités longues profitent de la connaissance des entités courtes, mais pas le contraire.
Parfois les mentions longues sont plus faciles à prédire que les courtes.

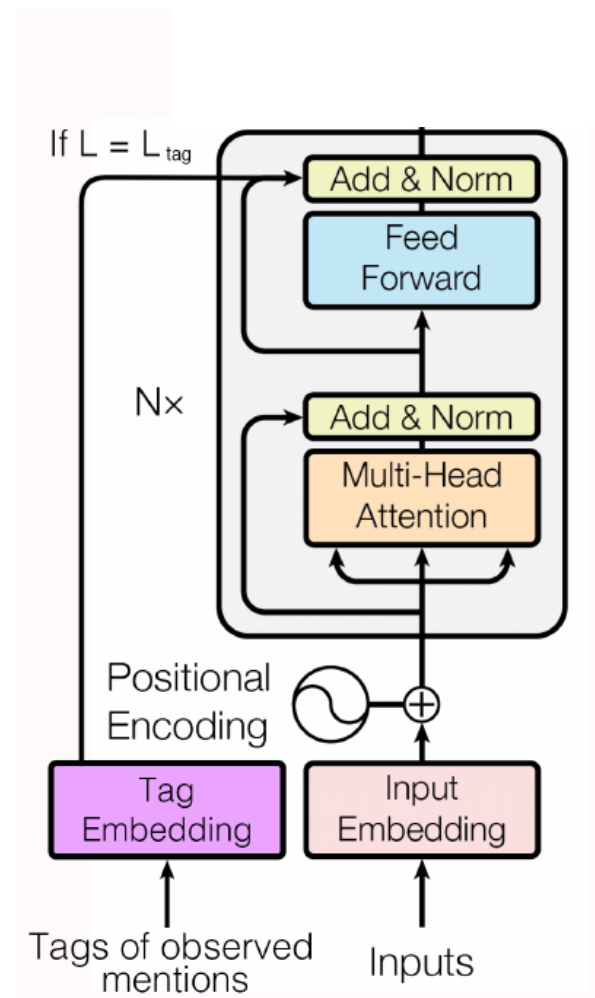
2. Modèle itératif sans contrainte

- Un transformer (Vaswani et al., 2017) prenant en entrée une séquence de mots



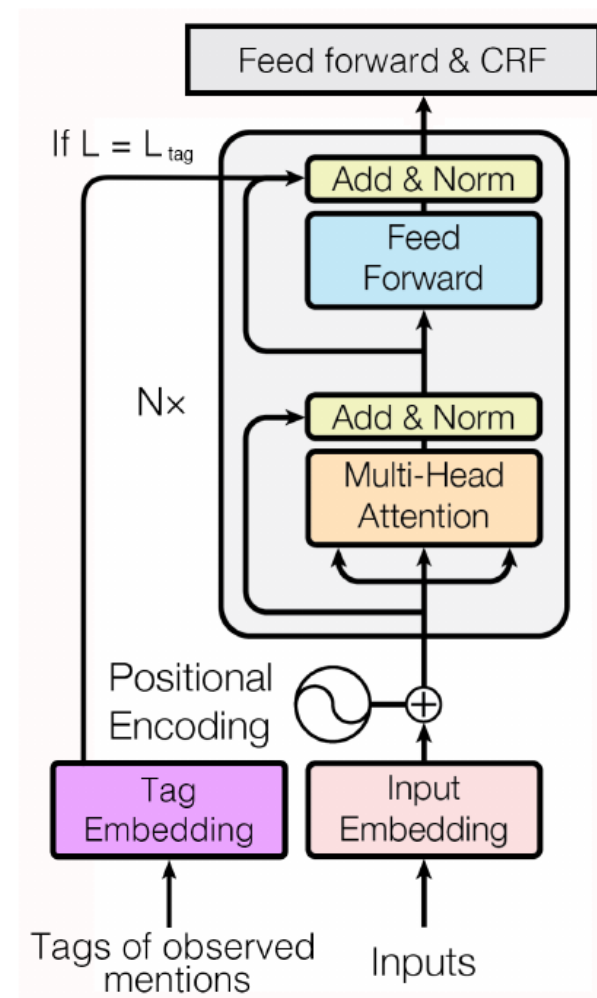
2. Modèle itératif sans contrainte

- Un transformer (Vaswani et al., 2017) prenant en entrée une séquence de mots ainsi qu'une liste de mentions déjà prédites à l'itération précédente (mentions converties en *embeddings*)



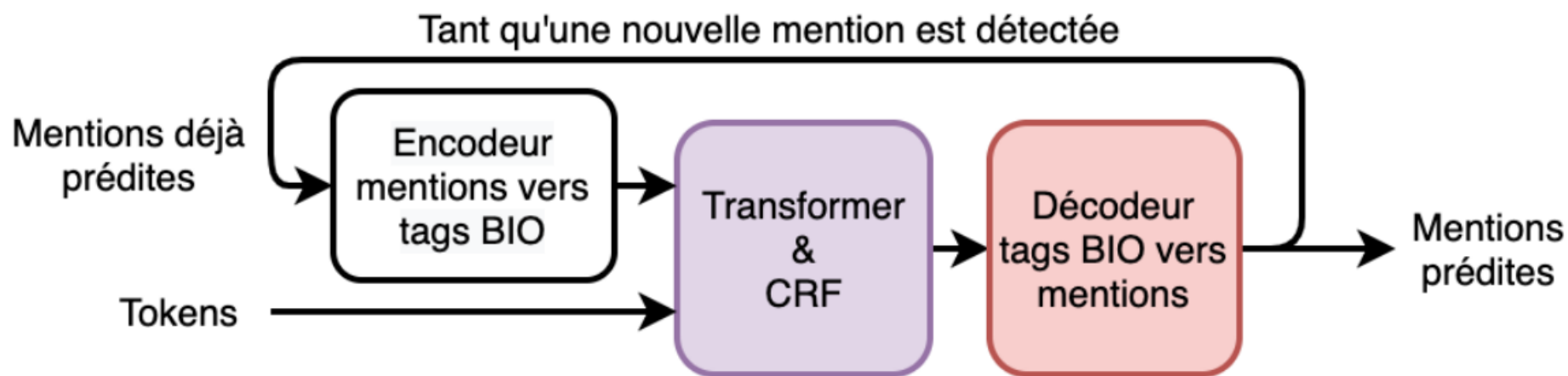
2. Modèle itératif sans contrainte

- Un transformer (Vaswani et al., 2017) prenant en entrée une séquence de mots ainsi qu'une liste de mentions déjà prédites à l'itération précédente (mentions converties en *embeddings*)
- Un CRF à la sortie



2. Modèle itératif sans contrainte

- Les mentions prédites à chaque itération ne se chevauchent pas, en revanche l'ensemble des mentions prédites à la fin des itérations peuvent se chevaucher.



2. Modèle itératif sans contrainte

- Même utilisation de CamemBERT + format BIO que pour le modèle précédent
- À l'inférence, pour chaque phrase, le modèle prédit la séquence de mentions la plus probable à partir d'une liste vide
- On arrête quand plus aucune mention n'est détectée.
- À l'entraînement, pour simuler cette extraction en plusieurs étapes, on sélectionne au hasard à chaque itération des mentions non recouvrantes, que l'on définira ensuite comme celles déjà extraites par le modèle
- Voir détails d'entraînement et hyper-paramètres dans l'article

2. Modèle itératif sans contrainte

- Dans un troisième run, on entraîne ce modèle avec 3 amorces aléatoires différentes et on fait voter ces trois modèles pour parvenir à la prédiction finale.

Résultat & discussion

Résultats

- Tâche 3.1, types « pathologie » et « signe ou symptôme » :
 - Run 1 (strates) : 0,35
 - Run 2 (itérations sans contrainte) : 0,59
 - Run 3 (itérations sans contrainte + vote) : 0,61
 - Meilleur : **0,66**
 - Médiane : 0,46
- Tâche 3.2, autres types :
 - Run 1 (strates) : 0,61
 - Run 2 (itérations sans contrainte) : 0,756
 - Run 3 (itérations sans contrainte + vote) : **0,763 (meilleur participant)**
 - Médiane : 0,62

(détails par type dans l'article)

Discussion

- L'analyse d'erreurs ne donne pas d'indices évident pour expliquer les performances plus faibles en 3.1 qu'en 3.2
- Nos modèles sont-ils moins performants ou bien le vainqueur de la 3.1 a-t-il utilisé des approches spécifiques ?
- Nous devons également réaliser de nouvelles expérimentations pour estimer si le fait de laisser la liberté au modèle dans l'ordre des combinaisons de mention, est bien ce qui conduit à des améliorations

Merci !

- Merci aux organisateurs pour le corpus et la mise en place du défi !