Modèles neuronaux hybrides pour la modélisation de séquences : le meilleur de trois mondes

Marco Dinarelli, Loïc Grobol

Modèles neuronaux hybrides pour la modélisation de séquences : le meilleur de trois mondes

Marco Dinarelli[1] Loïc Grobol[2,3]
(1) LIG, Bâtiment IMAG - 700 avenue Centrale - Domaine Universitaire de Saint-Martin-d'H'eres
(2) Lattice CNRS, 1 rue Maurice Arnoux, 92120 Montrouge, France
(3) ALMAnaCH Inria, 2 rue Simone Iff, 75589 Paris, France
marco.dinarelli@univ-grenoble-alpes.fr, loic.grobol@gmail.com

RÉSUMÉ

Nous proposons une architecture neuronale avec les caractéristiques principales des modèles neuronaux de ces dernières années : les réseaux neuronaux récurrents bidirectionnels, les modèles *encodeur-décodeur*, et le modèle *Transformer*. Nous évaluons nos modèles sur trois tâches d'étiquetage de séquence, avec des résultats aux environs de l'état de l'art et souvent meilleurs, montrant ainsi l'intérêt de cette architecture hybride pour ce type de tâches.

ABSTRACT

Hybrid Neural Networks for Sequence Modelling : The Best of Three Worlds

We propose a neural architecture with the main characteristics of the most successful neural models of the last years : bidirectional RNNs, *encoder-decoder*, and the *Transformer* model. Evaluation on three sequence labelling tasks yields results that are close to the state-of-the-art for all tasks and better than it for some of them, showing the pertinence of this hybrid architecture for this kind of tasks.

MOTS-CLÉS : Réseaux neuronaux, modélisation de séquences, MEDIA, WSJ, TIGER.

KEYWORDS : Neural Networks, sequence modelling, MEDIA, WSJ, TIGER.

1 Introduction

L'étiquetage de séquences est un problème important du TAL, de nombreux problèmes pouvant être modélisés comme des étiquetage de séquences. Les cas plus classiques sont l'étiquetage en parties du discours (*POS tagging*), la segmentation syntaxique, la reconnaissance d'entités nommées (Collobert *et al.*, 2011), ou encore la compréhension automatique de la parole dans les systèmes de dialogue humain-machine (De Mori *et al.*, 2008).

D'autres problèmes de TAL peuvent être divisés en plusieurs étapes, dont la première peut être modélisée comme étiquetage de séquences. Nous plaçons dans cette catégorie de problèmes l'analyse syntaxique, qui peut être décomposée en étiquetage en parties du discours et en analyse des constituants (Collins, 1997) ; la détection de chaînes de coréférences (Soon *et al.*, 2001 ; Ng & Cardie, 2002), qui peut être décomposée en détection de mentions et détection des mentions coréférentes ; mais aussi la détection d'entités nommées étendues (Grouin *et al.*, 2011 ; Dinarelli & Rosset, 2012a,b)

Plus largement, la traduction automatique et l'analyse syntaxique peuvent également être traitées comme des problèmes de prédiction de séquences bout-en-bout (Sutskever *et al.*, 2014 ; Bahdanau

et al., 2014 ; Vaswani *et al.*, 2017 ; Vinyals *et al.*, 2015), ainsi qu'une large classe de tâches de compréhension du langage (Devlin *et al.*, 2018). Il serait donc possible d'utiliser un modèle pour la prédiction de séquences dans un cadre d'apprentissage multi-tâche pour traiter la plupart des tâches de TAL, ce qui montre l'intérêt de la recherche d'architectures alternatives pour ce type de modèles

Les tâches en plusieurs étapes peuvent également être traitées de bout-en-bout par un modèle unique — comme l'analyse syntaxique en constituants, qui est typiquement traitée par un modèle qui effectue à la fois l'étiquetage en partie du discours et l'analyse syntaxique (Rush *et al.*, 2012). Cependant, même dans ce cas un pré-apprentissage de représentations par des réseaux neuronaux récurrents et leur ajustement sur des tâches plus simple — dont des tâches d'étiquetage de séquences — peut améliorer considérablement les performances (Peters *et al.*, 2018). On peut également rapprocher ce procédé le pré-apprentissage de plongements lexicaux, dont l'efficacité n'est plus à prouver (Lample *et al.*, 2016 ; Ma & Hovy, 2016).

Dans cet article nous nous limitons à proposer une architecture neuronale pour la modélisation de séquences dans un sens plus classique, c'est à dire l'étiquetage en partie du discours, l'analyse morpho-syntaxique, et l'étiquetage en concepts sémantiques tel qu'il est réalisé pour la tâche de compréhension de la parole dans le cadre du dialogue humain-machine (De Mori *et al.*, 2008).

En nous inspirant de (Chen *et al.*, 2018), duquel nous nous sommes inspirés également pour le titre de notre article, qui propose des modèles hybrides entre *Encodeur-décodeur* et *Transformer*, nous proposons une architecture avec les caractéristiques principales des modèles neuronaux plus efficaces proposés ces dernières années : les modèles RNNs bidirectionnels (Lample *et al.*, 2016 ; Ma & Hovy, 2016), les modèles *Encodeur-décodeur* (Sutskever *et al.*, 2014 ; Bahdanau *et al.*, 2014) et le modèle *Transformer* (Vaswani *et al.*, 2017).

Nous évaluons notre architecture sur trois tâches classiques d'étiquetage de séquences : la compréhension de la parole en français (corpus MEDIA) (Bonneau-Maynard *et al.*, 2006), le *POS tagging* en anglais (corpus WSJ) (Marcus *et al.*, 1993), et l'analyse morpho-syntaxique en allemand (corpus TIGER) (Brants *et al.*, 2004). Les résultats dépassent souvent l'état-de-l'art, et ils nous permettent de conclure dans tous les cas que notre architecture a sa place parmi les modèles neuronaux. L'adaptation de nos modèles à la traduction automatique et à l'analyse syntaxique est laissée comme travail futur.

2 Architectures Neuronales

Les modèles neuronaux proposés s'inspirent des modèles *LSTM+CRF* (Lample *et al.*, 2016 ; Ma & Hovy, 2016) pour encoder l'information en entrée ; des modèles *Encodeur-décodeur* pour l'architecture globale, et des modèles proposés dans (Dinarelli & Tellier, 2016a,b ; Dinarelli *et al.*, 2017 ; Dupont *et al.*, 2017) pour une prise de décision bidirectionnelle au niveau des unités de sortie (les étiquettes). À cette architecture nous avons ensuite ajouté certains caractéristiques du modèle *Transformer*.

2.1 Encodeur

L'encodeur de notre réseau est une couche cachée bidirectionnelle de type GRU (Cho *et al.*, 2014) qui prend en entrée une séquence S^{lex} représentant les mots par la concaténation de plongements de

mots non-contextuels ($E_w(w_i)$) et de représentations issues de leurs caractères $h_c(w_i)$. Celle-ci est calculée par un réseau de neurones récurrent dédié comme cela a été fait déjà dans la littérature Ma & Hovy (2016), la différence étant que nous utilisons une couche récurrente GRU au lieu d'une couche convolutionnelle.

La représentation des mots au niveau des caractères pour un mot quelconque w est calculée comme dans (Dinarelli & Grobol, 2019) :

$$\begin{aligned} S^c(w) &= (E_c(c_{w,1}), \ldots, E_c(c_{w,n})) \\ (h_c(c_{w,1}), \ldots, h_c(c_{w,n})) &= \text{GRU}_c(S^c(w), h_c^0) \\ h_c(w) &= \text{FFNN}(Sum(h_c(c_{w,1}), \ldots, h_c(c_{w,n}))) \end{aligned} \tag{1}$$

Avec les notations suivantes : E_c pour les plongements des caractères, $c_{w,i}$ pour le i-eme caractère du mot w, $S^c(w)$ pour la séquence de plongements des caractères du mot w, GRU_c pour la couche GRU pour les caractères, $h_c(c_{w,i})$ pour l'état caché associé au i-eme caractère du mot w. GRU_c, comme GRU_w, est une couche GRU bidirectionnelle.

La représentation cachée d'un mot w_i est ensuite calculée comme suit :

$$\begin{aligned} S^{lex} &= ([E_w(w_1), h_c(w_1)], \ldots, [E_w(w_N), h_c(w_N)]) \\ (h_{w_1}, \ldots, h_{w_N}) &= \text{GRU}_w(S^{lex}, h_w^0) \end{aligned} \tag{2}$$

Puisque la couche GRU_w parcourt la séquence en avant et en arrière, h_{w_i} dépend ainsi à la fois du mot w_i et de son contexte.Quand des traits additionnels sont disponibles en entrée, ils sont plongés de la même façon que les mots et concaténés à ces derniers dans la séquence S^{lex}.

2.2 Décodeurs

Notre modèle utilise une représentation des contextes d'étiquettes gauches et droits comme proposé par Dinarelli *et al.* (2017) ; Dupont *et al.* (2017). À la place des couches cachées linéaires nous utilisons cependant des couches récurrentes de type GRU. Nous utilisons une couche $\overleftarrow{\text{GRU}}_e$ *backward* pour encoder le contexte droit, et une couche $\overrightarrow{\text{GRU}}_e$ *forward* pour le contexte gauche. Ces couches prennent en entrée à la fois la représentation de l'information lexicale calculée par l'encodeur et les plongements des étiquettes $E_e(e_i)$, ce qui les rend similaires au décodeur utilisé dans l'architecture originale proposée par Sutskever *et al.* (2014) ; Bahdanau *et al.* (2014). Une évolution par rapport à cette architecture est notre utilisation de deux décodeurs, un pour le contexte droit et un pour le contexte gauche.

Le calcul du contexte droit par le décodeur *backward* $\overleftarrow{\text{GRU}}_e$ se fait formellement comme suit :

$$\overleftarrow{h_{e_i}} = \overleftarrow{\text{GRU}}_e([h_{w_i}, E_e(e_{i+1})], \overleftarrow{h_{e_{i+1}}}) \tag{3}$$

Le calcul du contexte gauche $\overrightarrow{h_{e_i}}$ est fait de façon similaire par le décodeur *forward* $\overrightarrow{\text{GRU}}_e$.

2.3 Couche de sortie

Afin que le modèle puisse prendre une décision globale, nous ajoutons une couche de sortie sur le décodeur *backward* composé d'une couche cachée linéaire suivie d'une fonction *log-softmax* qui calcule les log-probabilités des prédictions *backward* :

$$\begin{aligned} \text{log-P}(\overleftarrow{e_i}) &= \text{log-softmax}(W_{bw}[h_{w_i}, \overleftarrow{h_{e_i}}] + b_{bw}) \\ \overleftarrow{e_i} &= \text{argmax}(\text{log-P}(\overleftarrow{e_i})) \end{aligned} \tag{4}$$

Le décodeur *forward* prédit les étiquettes en utilisant à la fois les contextes d'étiquettes $\overrightarrow{h_{e_i}}$ et $\overleftarrow{h_{e_i}}$, ainsi que l'information lexicale h_{w_i} calculée par l'encodeur :

$$\begin{aligned} \text{log-P}(\overrightarrow{e_i}) &= \text{log-softmax}(W_o[\overrightarrow{h_{e_i}}, h_{w_i}, \overleftarrow{h_{e_i}}] + b_o) \\ \overrightarrow{e_i} &= \text{argmax}(\text{log-P}(\overrightarrow{e_i})) \end{aligned} \tag{5}$$

Pour renforcer le caractère global de la décision, la log-probabilité de la sortie finale est calculée comme la moyenne arithmétique des deux sorties *forward* et *backward* : $\frac{1}{2}(\text{log-P}(\overrightarrow{e_i}) + \text{log-P}(\overleftarrow{e_i}))$. [1]. Ceci permet d'inciter le modèle à fournir des prédictions de qualité dès la phase *backward* plutôt que de s'y contenter d'heuristiques grossières, voire de se reposer uniquement sur les prédictions de la phase *forward*.

2.4 Apprentissage

Tous nos modèles sont appris en minimisant l'opposé de la *log-vraisemblance* $\mathcal{LL}$ sur les données d'apprentissage. Formellement :

$$-\mathcal{LL}(\Theta|D) = -\sum_{d=1}^{|D|}\sum_{i=1}^{N_d} \frac{1}{2}(\text{log-P}(\overrightarrow{e_i}) + \text{log-P}(\overleftarrow{e_i})) + \frac{\lambda}{2}|\Theta|^2) \tag{6}$$

La première somme parcourt les données D de taille $|D|$, alors que la seconde somme parcourt chaque exemple d'apprentissage S_d, de longueur N_d.

Puisque les données utilisées dans ce travail ont une taille relativement petite, et nos modèles sont relativement complexes, nous ajoutons en terme de régularisation L_2 à la fonction de coût, dont λ constitue le coefficient.

2.5 Le meilleur de trois mondes

Le modèle décrit jusqu'ici reprend le principe introduit par (Dinarelli *et al.*, 2017 ; Dupont *et al.*, 2017) de prédire les étiquettes à partir d'une représentation des contextes gauche et droit aussi bien pour les mots que pour les étiquettes elle-même. Notre modèle réunit également les caractéristiques d'un RNN bidirectionnel, et du modèle *encodeur-décodeur*. Nous utilisons de plus deux décodeurs au lieu d'un seul comme dans l'architecture originale.

1. Ce qui équivaut au logarithme de la moyenne géométrique des probabilités

En partant de ce modèle nous avons ajouté certaines des caractéristiques du modèle *Transformer*.

Nous notons que l'article duquel nous nous sommes inspiré pour ce travail (Chen *et al.*, 2018), analysait la combinaison de deux architecture de type *encodeur-décodeur* , l'une récurrente (Sutskever *et al.*, 2014 ; Bahdanau *et al.*, 2014), l'autre basée sur le modèle *Transformer*, donc non-récurrente. Ce travail nous montre qu'une telle combinaison est possible mais aussi avantageuses par rapport aux deux architecture de départ.

Dans ce travail nous avons décidé de combiner trois architectures non pas pour des raisons de complémentarité, mais pour utiliser les forces de chacune d'entre elles, et pour pallier leur faiblesses. Dans les tâches sur lesquelles nous nous évaluons dans ce travail, il y a une correspondance un-à-un entre les unités d'entrée (les mots) et les unités de sortie (les étiquettes). Nos expériences préliminaires de détection de mentions pour la coréférence avec un *Transformer*, ainsi que certains résultats de la littérature (Guo *et al.*, 2019), suggère que se passer de cette correspondance conduit à des pertes remarquables de performance. [2] L'encodeur et les décodeurs de notre architecture utilisent pour cette raison l'information de correspondance un-à-un des tâches d'étiquetages de séquences. Nous sommes conscient cependant que cette information n'est pas utilisable dans des tâches comme la traduction automatique ou l'analyse syntaxique, auxquelles nous adapterons nos modèles dans le future.

Nous avons choisi d'utiliser une architecture *encodeur-décodeur* puisque nous avons montré que la prise en compte d'un contexte au niveau des étiquettes dans un décodeur est plus efficace qu'utiliser une couche CRF neuronale (Dinarelli *et al.*, 2017), ceci sera montré également dans ce travail. De plus, nous utilisons deux décodeurs, ce qui améliore encore les capacités de modélisation de contextes au niveau des unités de sortie. Cette solution n'est pas possible avec un *Transformer* classique.

Enfin, nous avons décidé d'intégrer certaines caractéristiques du *Transformer* pour pallier les limitations des RNNs concernant leur difficulté dans l'apprentissage, liée au problème de la longueur des parcours du signal d'apprentissage dans la phase de propagation en arrière (voir plus bas, mais aussi plus en détail (Vaswani *et al.*, 2017)).

Le modèle *Transformer* originel (Vaswani *et al.*, 2017) présentait une alternative aux réseaux récurrents fondée sur un mécanisme d'attention à têtes multiples (*Multi-Head Attention*). Des travaux récents (Dehghani *et al.*, 2018 ; Dai *et al.*, 2019) suggèrent cependant que ses performances peuvent être améliorées par l'ajout de certaines formes de récurrence.

Une autre caractéristique intéressante du *Transformer* est l'utilisation des connexions résiduelles (*Skip connections*). Celles-ci permettent de mitiger le problème classique d'évanouissement du gradient dans les réseaux neuronaux profonds. Dans le cas des RNN, l'utilisation de mécanismes de portes permet déjà de limiter ce phénomène, mais nos expériences suggèrent que pour des séquences très longues, elle ne suffit pas à s'en affranchir complètement.

Le mécanisme d'attention ne présente pas ce problème. Chaque élément de la séquence en entrée étant reliée à un nombre fixe de couches, le parcours de rétro-propagation du signal d'apprentissage est assez court par rapport aux RNNs. De plus, l'utilisation des connexions résiduelles renforce davantage la puissance du signal rétro-propagé, permettant à celui-ci de sauter une couche, et donc son affaiblissement, à chaque fois qu'une connexion résiduelle est utilisée.

Nous pouvons interpréter chaque bloc d'une architecture *Transformer*, que ce soit l'encodeur ou le décodeur, comme étant composé par un sous-module qui *"encode"* des traits contextuels, que nous appelons avec abus de langage *Encoder*, et par un sous-module *feed-forward* qui *"transforme"* ces

2. Nous obtenons plus que 3 points de F-mesure en moins avec le *Transformer*

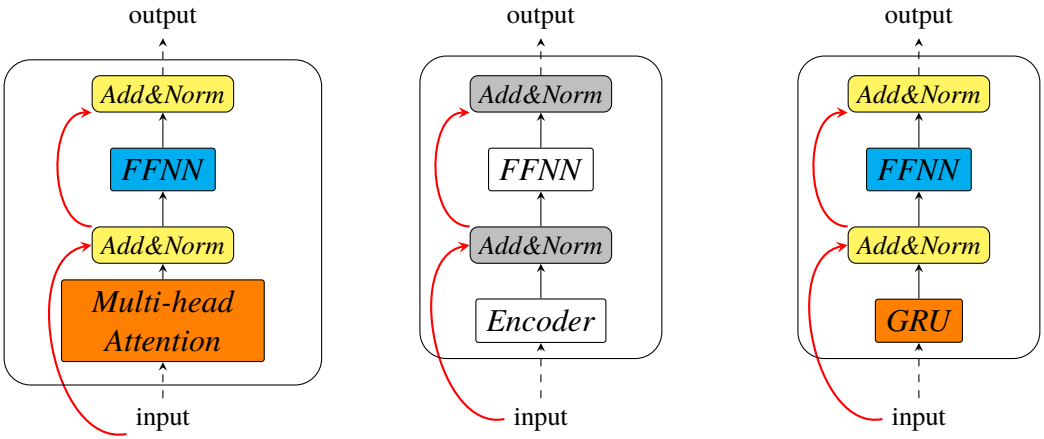

FIGURE 1 – Architecture générale, d'un point de vue conceptuel, de chaque bloc d'un réseau *Transformer* (au centre), sa réalisation pratique dans un modèle *Transformer* (à gauche) et sa réalisation dans notre réseau (à droite), dans lequel nous utilisons une couche GRU pour encoder des traits contextuels

traits en les plongeant dans un espace de traits "profonds". Dans l'architecture *Transformer* (Vaswani *et al.*, 2017) la sortie des deux sous-modules est additionnée à l'entrée du sous-module (*skip connection*) et normalisée au niveau de la couche (*layer normalisation*). Cette interprétation est montrée dans la figure 1 au centre. Les connexions résiduelles sont montrées en rouge.

Le *Transformer* présenté par Vaswani *et al.* (2017) instancie l'architecture générique décrite ci-dessus en utilisant comme *Encoder* le mécanisme d'attention à têtes multiple. Cette architecture est présentée à gauche dans la figure 1.

Nous avons modifié notre architecture neuronale pour qu'elle instancie également l'architecture générique décrite plus haut. Dans notre architecture nous utilisons une couche récurrente GRU comme *Encoder*, et exactement le même réseau *Feed-Forward* que dans l'architecture proposée par Vaswani *et al.* (2017), c'est à dire avec deux couches. Notre architecture est montrée à droite dans la figure 1.

Les caractéristiques du modèle *Transformer* que nous intégrons dans notre modèle sont donc les *skip-connections* et la normalisation au niveau des couches cachées (*Add&Norm* dans la figure 1), ainsi que le réseau *Feed-Forward* qui re-encode la sortie des couches GRU (*FFNN* dans la figure 1). Ainsi, en suivant la même chaîne d'opérations suggérée par Chen *et al.* (2018), la sortie de l'encodeur h_{w_i} est calculée comme suit :

$$\begin{aligned} \hat{S}_i^{lex} &= \text{Norm}(S_i^{lex}) \\ \hat{h}_{w_i} &= GRU_w(\hat{S}_i^{lex}, h_{w_{i-1}}) \\ h_{w_i} &= \text{FFNN}(\text{Norm}(\text{Dropout}(\hat{h}_{w_i}) + \hat{S}_i^{lex})) \end{aligned} \tag{7}$$

où nous avons indiqué avec Norm la normalisation des couches (*Layer Normalisation*) et avec *Dropout* la régularisation *dropout* (Srivastava *et al.*, 2014). Les autres couches GRU introduites plus haut sont modifiées de façon similaire.

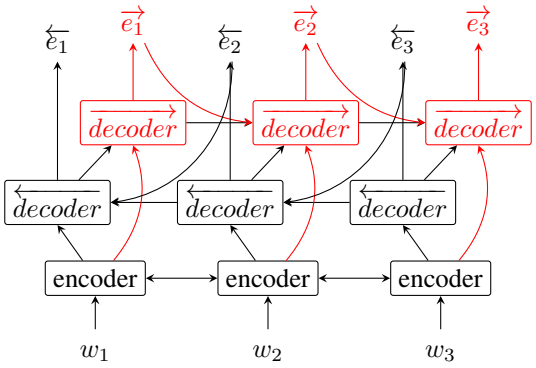

FIGURE 2 – Schéma de notre architecture neuronale, équivalente d'un point de vue global à celle de Dinarelli & Grobol (2019)

Nous ne reprenons pas ici le mécanisme d'attentions multiples des *Transformer*, la prise en compte d'un contexte pertinent autour du mot à étiqueter reposant sur l'utilisation de la couche GRU_w (cf. section 2.1). Ce choix est motivé entre autre par les conclusions de Levy *et al.* (2018), qui montrent que les réseaux récurrents, dont GRU, encodent un long contexte sous forme de moyennes pondérée de toutes les entrées, constituant ainsi une forme d'attention. L'utilisation de couches GRU nous permet également de nous passer des *plongements positionnels* utilisés par Vaswani *et al.* (2017), la structure séquentielle des entrées étant implicitement encodée par les couches récurrentes GRU. [3]

L'architecture globale de notre réseau final peut être décrite par le même schéma utilisé que pour (Dinarelli & Grobol, 2019) (cf figure 2), les différences étant dans les calculs des couches décrits dans les sections précédentes.

3 Évaluation

Nous évaluons nos modèles sur trois tâches :

Le corpus français MEDIA (Bonneau-Maynard *et al.*, 2006) a été créé pour l'évaluation de systèmes de dialogues destinés à fournir des informations touristiques sur les hôtels en France. Les données ont été annotées manuellement suivant une ontologie de concepts riche. Des composants sémantiques peuvent être combinés pour former des étiquettes sémantiques complexes. [4] Les propriétés statistiques des données d'apprentissage, de développement et de test du corpus MEDIA sont données dans le tableau 1.

La tâche MEDIA peut être modélisée comme un étiquetage en concepts sémantiques de séquences en utilisant la convention *BIO* (Ramshaw & Marcus, 1995). Nous nous sommes déjà évalué sur cette tâche dans le passé, en utilisant une variété assez large de modèle probabilistes (Dinarelli *et al.*, 2009a,b ; Quarteroni *et al.*, 2009 ; Hahn *et al.*, 2010 ; Dinarelli, 2010 ; Dinarelli & Rosset, 2011 ;

3. Nous n'excluons pas cependant que l'utilisation des *plongements positionnels* puisse améliorer davantage notre architecture

4. Par exemple l'étiquette `localisation` peut être combinée avec les composants `ville`, `distance-relative`, `localisation-relative-générale`, `rue`, etc.

	Training		Validation		Test	
# phrases	12 908		1 259		3 005	
	Mots	Concepts	Mots	Concepts	Mots	Concepts
# Mots	94 466	43 078	10 849	4 705	25 606	11 383
dictionnaire	2 210	99	838	66	1 276	78
OOV%	–	–	1,33	0,02	1,39	0,04

TABLE 1 – Statistiques des données du corpus français MEDIA

	Training		Validation		Test	
# phrases	38 219		5 527		5 462	
	Mots	Étiquettes	Mots	Étiquettes	Mots	Étiquettes
# Mots	912 344	–	131 768	–	129 654	–
dictionnaire	43 210	45	15 081	45	13 968	45
OOV%	–	–	22,61	0	20,00	0

TABLE 2 – Statistiques des données du corpus anglais *WSJ*

Dinarelli *et al.*, 2011).

Pour la tâche MEDIA des classes de mots sont disponibles pour permettre aux modèles une meilleurs généralisation sur certaines mots appartenant à des catégories dont des listes peuvent être facilement récupérées dans le cadre d'une application d'interaction humain-machine comme les systèmes de dialogue. Des exemples de ces classes sont les noms des villes en France (classe *VILLE*), les noms des marques d'hôtel (*HOTEL*), les quantités correspondant à des montant (*MONTANT*), etc. Nous avons utilisés ces classes pour certaines expériences, ceci est indiqué par *FEAT* (pour *features*) dans les tableaux de la section suivante.

Le corpus anglais Penn TreeBank (Marcus *et al.*, 1993), indiqué avec WSJ dans la suite, constitue l'une des tâches les plus utilisées pour l'évaluation de modèles pour l'étiquetage de séquence. La tâche consiste à associer chaque mot avec son étiquette (*POS tag*). Nous utilisons la répartition habituelle des données : les sections 0-18 pour l'apprentissage, les sections 19-21 comme données de développement et les sections 22-24 comme données de test. Les propriétés statistiques des données d'apprentissage, de développement et de test du corpus WSJ sont données dans le tableau 2.

Le corpus allemand TIGER (Brants *et al.*, 2004) est annoté avec des informations morpho-syntaxiques riches, comprenant non seulement les *POS* comme dans le corpus WSJ, mais aussi le genre, le nombre, le cas et des informations de conjugaison pour les verbes. Cette tâche est proche du *POS tagging* d'un point de vue de modélisation, avec une plus grande difficulté liée non seulement à la langue, mais aussi au nombre assez grand d'étiquettes que le modèle doit désambiguïser (694 au total, contre 138 dans MEDIA et 45 dans le WSJ). Nous utilisons la même répartition de donnée que (Lavergne & Yvon, 2017). Les propriétés statistiques des données d'apprentissage, de développement et de test de ce corpus sont données dans le tableau 3. Comme nous pouvons le constater la taille des dictionnaires, que ce soit pour les mots ou pour les étiquettes, est assez importante.

Nous notons que le taux de mots hors vocabulaire (OOV) dans le corpus MEDIA est assez faible. Il y a en effet uniquement deux mots hors vocabulaire dans les données de développement et de test, il s'agit en plus de mots vides, il n'y a donc pratiquement pas de mots inconnus.

En revanche le taux de mots hors vocabulaire dans les corpus WSJ et TIGER est assez élevé : environ 1 sur 5 dans le premier et 1 sur 3 dans le second.

	Training		Validation		Test	
# phrases	40 472		5 000		5 000	
	Mots	Étiquettes	Mots	Étiquettes	Mots	Étiquettes
# Mots	719 530	–	76 704	–	92 004	–
dictionnaire	77 220	681	15 852	501	20 149	537
OOV%	–	–	30,90	0,01	37,18	0,015

TABLE 3 – Statistiques des données du corpus allemand TIGER

Modèle	**Précision**	**F1**	**CER**
MEDIA DEV			
GRU+LD-RNN	89.11	85.59	11.46
GRU+LD-RNN$_{le}$	89.42	86.09	10.58
GRU+LD-RNN$_{le}$ seg-len 15	**89.97**	**86.57**	**10.42**
fw-GRU+LD-RNN$_{le}$ seg-len 15	89.51	85.94	11.40

TABLE 4 – Comparaison des résultats sur les données de developpement du corpus MEDIA, sans et avec l'information lexicale au niveau des décodeurs $\overleftarrow{\text{GRU}}_e$ and $\overleftarrow{\text{GRU}}_e$ (Seq2Biseq$_{le}$ dans le tableau)

3.1 Réglages

Pour le développement de nos modèles nous avons utilisé le corpus MEDIA, le corpus plus petit et permettant donc une optimisation plus rapide. Nos réglages sont les mêmes que dans (Dinarelli *et al.*, 2017 ; Dinarelli & Grobol, 2019), qui utilisait également ces données pour l'évaluation. Les réglages sur le corpus WSJ sont les mêmes, sauf pour les plongements des mots (300 dimensions) et le taux d'apprentissage ($2{,}5 \times 10^{-4}$). Nous utilisons ces mêmes réglages pour WSJ et TIGER.

Comme nous l'avons expliqué dans (Dinarelli & Grobol, 2019), dans un premier temps nos modèles ne passaient pas dans la mémoire des nos GPU. Pour resoudre ce problème nous avons utilisé deux solutions. La première consiste en organiser les données d'apprentissage comme un seul flux de tokens. Ce flux est ensuite découpé en segments de taille fixe qui se chevauchent, avec un glissement d'un token entre deux segments consécutifs. La seconde solution est plus classique et consiste en regrouper ensemble les phrases de la même longueur. Ceci crée des groupes petits pour des phrases de grande taille, qui sont plus rares, et des grands groupes pour des phrases de taille petite et moyenne. Dans les deux cas les groupes passent en mémoire sans problème.

Dans nos expériences nous avons trouvé que la première solution marche bien mieux pour MEDIA, alors que sur les données *WSJ* et *TIGER* les deux solutions sont à peu près équivalentes. Pour des données de cette taille nous préférons donc la seconde solution, qui est plus intuitive et générale.

3.2 Résultats

Les résultats sur la tâche MEDIA sont des moyennes sur 10 expériences, les paramètres entraînables sont réinitialisés aléatoirement [5] pour chacune d'entre elles. Sur cette tâche nous nous évaluons en termes de précision et, puisqu'il faut reconstituer les concepts à partir des étiquettes BIO, aussi avec la F-mesure et le *Concept Error Rate* (CER). Le CER est calculé en alignant la prédiction avec l'annotation de référence, et en divisant ensuite la somme des insertions, substitutions et délétions par

5. Suivant une distribution uniforme pour les couches GRU et suivant (He *et al.*, 2015) pour les couches linéaires.

Modèle	Précision	F1	CER
MEDIA DEV			
GRU+LD-RNN	89.97	86.57	10.42
GRU+LD-RNN$_{\text{2-opt}}$	**90.22**	86.88	9.97
GRU+LD-RNN+FEAT$_{\text{2-opt}}$	90.14	**87.05**	**9.54**
MEDIA TEST			
BiGRU+CRF (Dinarelli *et al.*, 2017)	–	86.69	10.13
LD-RNN$_{\text{deep}}$ (Dinarelli *et al.*, 2017)	–	87.36	9.8
GRU+LD-RNN	89.57	87.50	10.26
GRU+LD-RNN$_{\text{2-opt}}$	89.79	87.69	9.93
GRU+LD-RNN+FEAT$_{\text{2-opt}}$	**90.12**	**87.94**	**9.48**

TABLE 5 – Performances des différentes variantes de notre architecture pour la tâche d'étiquetage sémantique sur MEDIA, comparées à l'état de l'art

le nombre de concepts de la référence. Sur les autres tâches, dans lesquelles à chaque mot correspond une étiquette, nous nous évaluons uniquement en termes de précision.

Les résultats présentés dans le tableau 4 sont les mêmes discutés dans notre précèdent travail (Dinarelli & Grobol, 2019), nous les ré-discutons également ici pour fournir un travail autonome et complet. Dans ces expériences nous avons voulu tester les capacités des décodeurs à construire une représentation efficace du contexte au niveau des étiquettes, et à filtrer des informations bruitées ou non-pertinentes dans la construction de telles représentations. Pour tester cela nous avons effectué des expériences sans (GRU+LD-RNN) et avec (GRU+LD-RNN$_{le}$) l'information lexicale h_{w_i} au niveau des décodeurs. Comme nous pouvons le constater dans le tableau 4 le modèle utilisant l'information lexicale obtient des résultats significativement meilleurs. Ceci, en prenant en compte que les deux modèles GRU+LD-RNN et GRU+LD-RNN$_{le}$ utilisent tous les deux l'information lexicale aussi au niveau de la couche de sortie (cf. section 2.3), prouve la capacité des décodeurs à construire une représentation plus efficace du contexte d'étiquettes en partant d'informations en entrée plus riches.

Pour tester la capacité à filtrer des informations non-pertinentes, nous avons effectué des expériences en variant la taille des segments de 10 (par défaut) à 15 tokens (cf section 3.1). À nouveau nous pouvons constater que les résultats s'améliorent avec des segments de taille 15 (GRU+LD-RNN$_{le}$ seg-len 15 dans le tableau 4). Compte tenu que MEDIA est constitué de transcriptions de l'orale, donc de données bruitées, ces améliorations montrent une bonne capacité de filtre des informations non-pertinentes par les décodeurs.

La dernière ligne du tableau 4 montre les résultats obtenus en n'utilisant que le décodeur *forward* (fw-GRU+LD-RNN$_{le}$ seg-len 15 dans le tableau). Ces résultats prouvent tout l'intérêt à utiliser à la fois un contexte gauche et un contexte droit au niveau des étiquettes, le modèle employant deux décodeurs étant largement meilleurs que celui en utilisant un seul.

Puisque l'utilisation de l'information lexicale h_{w_i} dans les décodeurs, et les segments de taille 15 mènent aux meilleurs résultats, ces réglages sont choisi par défaut et par la suite ils ne seront pas specifiés à côté du nom du modèle, qui sera simplement GRU+LD-RNN. Pour rappel, l'organisation des données en segment est utilisé uniquement pour MEDIA, pour les autres tâches nous utilisons une organisation en groupe de phrases de la même taille.

Modèle	Précision	
	WSJ DEV	WSJ TEST
LD-RNN$_{\text{deep}}$	96.90	96.91
LSTM+CRF (Ma & Hovy, 2016)	–	97.13
GRU+LD-RNN	97.13	97.20
GRU+LD-RNN$_{2-opt}$	**97.22**	**97.36**
LSTM+CRF + Glove (Ma & Hovy, 2016)	97.46	97.55
LSTM+LD-RNN + Glove (Zhang *et al.*, 2018)	–	97.59

TABLE 6 – Performances des différentes variantes de notre architecture pour la tâche de POS-tagging sur WSJ, comparées à l'état de l'art

Les résultats complet sur MEDIA sont donnés dans le tableau 5. Dans ce tableau nous montrons les résultats sur les données de développement et de test. Sur ces dernières nous nous comparons avec nos résultats précédents (Dinarelli *et al.*, 2017). Dans un premier temps nous avons entraîné un modèle sans les classes de mots disponibles pour cette tâche (GRU+LD-RNN, cf le début de la section 3), et sans utiliser les fonctionnalités d'un modèle *Transformer*. Alors que nous améliorions légèrement l'état-de-l'art en termes de F-mesure (87,50 contre 87,36), notre CER restait supérieur.

En analysant les sorties de nos modèles sur les données de développement nous avons remarqué des signes clairs indiquant que le modèle ignorait l'information donnée par le contexte droit au niveau des étiquettes. Nous en avons conclus que notre modèle souffrait du même problème mentionné dans (Vaswani *et al.*, 2017) pour les RNNs. Nous notons que ce problème, bien qu'il a été remarqué sur des données particulières, concerne une limitation du modèle sur sa capacité à prendre en compte le contexte droit, il s'agit donc d'un comportement générale et non lié à ce jeu de données spécifiques. Comme nous le verrons pas la suite, en effet les modifications mises en place pour résoudre ce problème se sont avérées assez efficaces sur toutes les tâches sur lesquelles nous nous évaluons.

Nous avons alors ajouté à notre architecture les fonctionnalités du modèle *Transformer* mentionnées dans la section 2.5. Puisque notre modèle utilise deux décodeurs, nous avons aussi entraîné le système avec 2 optimiseurs (GRU+LD-RNN$_{\text{2-opt}}$), chacun minimisant la log-vraisemblance de la sortie de chaque décodeur. Les résultats obtenus avec ce modèle dépassent tous les précédents pour toutes les mesures d'évaluation, et dépassent également l'état-de-l'art dans les mêmes conditions.

En ajoutant les classes des mots disponibles dans MEDIA (GRU+LD-RNN+FEAT$_{\text{2-opt}}$), les résultats s'améliorent encore pour toutes les mesures d'évaluation.

Étant donné leur taille réduite, nous avons utilisé les données MEDIA pour une optimisation plus rapide des choix au niveau de l'architecture et de la plupart des hyper-paramètres. Les seuls paramètres que nous avons ré-optimisé sur le corpus WSJ sont le taux d'apprentissage et la taille de la couche cachée (cf. section 3.1). La taille des plongements des mots a été choisie en se basant sur (Zhang *et al.*, 2018). Nous avons effectué une seule expérience sur le corpus TIGER. Puisque celle-ci a donné des bons résultats, nous n'avons pas optimisé davantage le modèle.

Les résultats sur le *POS tagging* du corpus WSJ sont montrés dans le tableau 6. Dans ce cas également l'utilisation des fonctionnalités d'un modèle *Transformer* donne des améliorations sur la précision. Nous notons que notre modèle, que ce soit avec un seul optimiseur ou deux et avec en plus les fonctionnalités du *Transformer*, améliore le modèle *LSTM+CRF* (Ma & Hovy, 2016) sans utiliser des

Modèle	Précision	
	TIGER DEV	TIGER TEST
GRU+LD-RNN$_{2-opt}$	**93.90** (98.30)	**91.86** (97.74)
VO-CRF (Lavergne & Yvon, 2017)	–	88.78

TABLE 7 – Performances des différentes variantes de notre architecture pour l'étiquetage morpho-syntaxique sur TIGER, comparées à l'état de l'art

plongements pré-appris avec *GloVe* (Pennington *et al.*, 2014). À titre de comparaison nous montrons aussi les meilleurs résultats de la littérature sur cette tâche. Bien que nos résultats ne dépassent pas l'état-de-l'art, ils en restent assez proche. Nous notons cependant que nos premières expériences avec des plongements pré-appris n'améliorent pas l'état-de-l'art [6]. Des analyses sont en cours pour comprendre ce manque de gain.

Dans le tableau 7 nous montrons les résultats obtenus sur la tâche d'étiquetage morpho-syntaxique de l'allemand (TIGER). À notre connaissance le meilleur résultat de la littérature est celui obtenu par Lavergne & Yvon (2017) avec un CRF d'ordre variable (VO-CRF). Nous pouvons donc constater que notre modèle améliore l'état-de-l'art sur cette tâche. Entre parenthèses nous montrons également les résultats du *POS tagging*. Ces résultats sont obtenus des précédents en ne considérant que l'étiquette POS, sans apprentissage spécifique.

4 Conclusions

Dans cet article nous avons proposé un modèle neuronal pour la modélisation de séquences qui réunit des caractéristiques des modèles neuronaux plus populaires de ces dernières années : les RNNs bi-directionnels, les modèles *encodeur-décodeur* et le modèle *Transformer*. Une évaluation sur trois tâches classiques d'étiquetage de séquences montre que notre modèle est très efficace pour ce type de problèmes. En effet il obtient souvent des résultats à l'état-de-l'art, et il en est proche dans tous les cas.

Remerciements

Cette recherche s'insère dans le programme « Investissements d'Avenir » géré par l'Agence Nationale de la Recherche ANR-10-LABX-0083 (Labex EFL).

Ce travail a par ailleurs bénéficié du soutien de l'ANR DEMOCRAT (Description et modélisation des chaînes de référence : outils pour l'annotation de corpus et le traitement automatique), projet ANR-15-CE38-0008.

6. Contrairement au modèle *LSTM+CRF* (Ma & Hovy, 2016) dont la précision est grandement améliorée par les plongements pré-appris *GloVe*.

Références

Bahdanau D., Cho K. & Bengio Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, **abs/1409.0473**.

Bonneau-Maynard H., Ayache C., Bechet F., Denis A., Kuhn A., Lefèvre F., Mostefa D., Qugnard M., Rosset S. & Servan, S. Vilaneau J. (2006). Results of the french evalda-media evaluation campaign for literal understanding. In *LREC*, p. 2054–2059, Genoa, Italy.

Brants S., Dipper S., Eisenberg P., Hansen-Schirra S., Konig E., Lezius W., Rohrer C., Smith G. & Uszkoreit H. (2004). TIGER : Linguistic interpretation of a german corpus. *Research on Language and Computation*, **2**(4), 597–620.

Chen M. X., Firat O., Bapna A., Johnson M., Macherey W., Foster G., Jones L., Schuster M., Shazeer N., Parmar N., Vaswani A., Uszkoreit J., Kaiser L., Chen Z., Wu Y. & Hughes M. (2018). The Best of Both Worlds : Combining Recent Advances in Neural Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, p. 76–86 : Association for Computational Linguistics.

Cho K., van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H. & Bengio Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1724–1734 : Association for Computational Linguistics.

Collins M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL*, p. 16–23, Stroudsburg, PA, USA : Association for Computational Linguistics.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K. & Kuksa P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, **12**.

Dai Z., Yang Z., Yang Y., Carbonell J., Le Q. V. & Salakhutdinov R. (2019). Transformer-XL : Attentive Language Models Beyond a Fixed-Length Context. *arXiv preprint 1901.02860*.

De Mori R., Bechet F., Hakkani-Tur D., McTear M., Riccardi G. & Tur G. (2008). Spoken language understanding : A survey. *IEEE Signal Processing Magazine*, **25**, 50–58.

Dehghani M., Gouws S., Vinyals O., Uszkoreit J. & Kaiser L. (2018). Universal transformers. *CoRR*, **abs/1807.03819**.

Devlin J., Chang M.-W., Lee K. & Toutanova K. (2018). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint 1810.04805*.

Dinarelli M. (2010). *Spoken Language Understanding : from Spoken Utterances to Semantic Structures*. PhD thesis, International Doctoral School in Information and Communication Technology, Dipartimento di Ingegneria e Scienza dell' Informazione, via Sommarive 14, 38100 Povo di Trento (TN), Italy.

Dinarelli M. & Grobol L. (2019). Seq2biseq : Bidirectional output-wise recurrent neural networks for sequence modelling. *CoRR*, **abs/1904.04733**.

Dinarelli M., Moschitti A. & Riccardi G. (2009a). Concept segmentation and labeling for conversational speech. In *Proceedings of the International Conference of the Speech Communication Assosiation (Interspeech)*, Brighton, U.K.

Dinarelli M., Moschitti A. & Riccardi G. (2009b). Re-ranking models based on small training data for spoken language understanding. In *Conference of Empirical Methods for Natural Language Processing*, p. 11–18, Singapore.

Dinarelli M., Moschitti A. & Riccardi G. (2011). Discriminative reranking for spoken language understanding. *IEEE TASLP*, **20**, 526–539.

Dinarelli M. & Rosset S. (2011). Hypotheses selection criteria in a reranking framework for spoken language understanding. In *Conference of Empirical Methods for Natural Language Processing*, p. 1104–1115, Edinburgh, U.K.

Dinarelli M. & Rosset S. (2012a). Tree representations in probabilistic models for extended named entity detection. In *European Chapter of the Association for Computational Linguistics (EACL)*, p. 174–184, Avignon, France.

Dinarelli M. & Rosset S. (2012b). Tree-structured named entity recognition on ocr data : Analysis, processing and results. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk & S. Piperidis, Eds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey : European Language Resources Association (ELRA).

Dinarelli M. & Tellier I. (2016a). Improving recurrent neural networks for sequence labelling. *CoRR*, **abs/1606.02555**.

Dinarelli M. & Tellier I. (2016b). New recurrent neural network variants for sequence labeling. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics*, Konya, Turkey : Lecture Notes in Computer Science (Springer).

Dinarelli M., Vukotic V. & Raymond C. (2017). Label-dependency coding in Simple Recurrent Networks for Spoken Language Understanding. In *Interspeech*, Stockholm, Sweden.

Dupont Y., Dinarelli M. & Tellier I. (2017). Label-dependencies aware recurrent neural networks. In *Proceedings of CICling*, Budapest, Hungary : LNCS, Springer.

Grouin C., Dinarelli M., Rosset S., Wisniewski G. & Zweigenbaum P. (2011). Coreference resolution in clinical reports. the limsi participation in the i2b2/va 2011 challenge. In *In Proceedings of i2b2/VA 2011 Coreference Resolution Workshop*.

Guo Q., Qiu X., Liu P., Shao Y., Xue X. & Zhang Z. (2019). Star-transformer. *CoRR*, **abs/1902.09113**.

Hahn S., Dinarelli M., Raymond C., Lefèvre F., Lehen P., De Mori R., Moschitti A., Ney H. & Riccardi G. (2010). Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE TASLP*, **99**.

He K., Zhang X., Ren S. & Sun J. (2015). Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, p. 1026–1034.

Lample G., Ballesteros M., Subramanian S., Kawakami K. & Dyer C. (2016). Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 260–270 : Association for Computational Linguistics.

Lavergne T. & Yvon F. (2017). Learning the structure of variable-order crfs : a finite-state perspective. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 433–439 : Association for Computational Linguistics.

Levy O., Lee K., FitzGerald N. & Zettlemoyer L. (2018). Long short-term memory as a dynamically computed element-wise weighted sum. In *Proceedings of ACL*, p. 732–739 : ACL.

Ma X. & Hovy E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of ACL*.

Marcus M. P., Santorini B. & Marcinkiewicz M. A. (1993). Building a large annotated corpus of english : The penn treebank. *COMPUTATIONAL LINGUISTICS*, **19**(2).

Ng V. & Cardie C. (2002). Improving Machine Learning Approcahes to Corefrence Resolution. In *Proceedings of ACL'02*, p. 104–111.

Pennington J., Socher R. & Manning C. D. (2014). Glove : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543.

Peters M., Neumann M., Iyyer M., Gardner M., Clark C., Lee K. & Zettlemoyer L. (2018). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, volume 1, p. 2227–2237 : Association for Computational Linguistics.

Quarteroni S., Riccardi G. & Dinarelli M. (2009). What's in an ontology for spoken language understanding. In *Proceedings of the International Conference of the Speech Communication Assosiation (Interspeech)*, Brighton, U.K.

Ramshaw L. & Marcus M. (1995). Text chunking using transformation-based learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*, p. 84–94, Cambridge, MA, USA.

Rush A. M., Reichart R., Collins M. & Globerson A. (2012). Improved parsing and pos tagging using inter-sentence consistency constraints. In *Proceedings of EMNLP-CoNLL*, Stroudsburg, PA, USA.

Soon W. M., Ng H. T. & Lim D. C. Y. (2001). A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, **27**(4), 521–544.

Srivastava N., Hinton G., Krizhevsky A., Sutskever I. & Salakhutdinov R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**, 1929–1958.

Sutskever I., Vinyals O. & Le Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, Cambridge, MA, USA : MIT Press.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L. & Polosukhin I. (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett, Eds., *Advances in Neural Information Processing Systems 30*, p. 5998–6008. Curran Associates, Inc.

Vinyals O., Kaiser L., Koo T., Petrov S., Sutskever I. & Hinton G. (2015). Grammar As a Foreign Language. In *Proceedings of the 28th International Conference on Neural Information Processing*, volume 2 of *NIPS'15*, p. 2773–2781, Cambridge, MA, USA : MIT Press.

Zhang Y., Chen H., Zhao Y., Liu Q. & Yin D. (2018). Learning tag dependencies for sequence tagging. In *International Joint Conference on Artificial Intelligence (IJCAI)*.