Modélisation d'un contexte global d'étiquettes pour l'étiquetage de séquences dans les réseaux neuronaux récurrents

Marco Dinarelli[1] Loïc Grobol[1, 2]
(1) Lattice, CNRS, ENS, Université Sorbonne Nouvelle, PSL, USPC, 1 rue Maurice Arnoux, 92120 Montrouge, France
(2) ALMAnaCH, Inria, 2 rue Simone Iff, 75589 Paris, France
marco.dinarelli@ens.fr, loic.grobol@ens.fr

RÉSUMÉ

Depuis quelques années, les réseaux neuronaux récurrents ont atteint des performances à l'état-de-l'art sur la plupart des problèmes de traitement de séquences. Notamment les modèles *sequence to sequence* et les CRF neuronaux se sont montrés particulièrement efficaces pour ce genre de problèmes. Dans cet article, nous proposons un réseau neuronal alternatif pour le même type de problèmes, basé sur l'utilisation de plongements d'étiquettes et sur des réseaux *à mémoire*, qui permettent la prise en compte de contextes arbitrairement longs. Nous comparons nos modèles avec la littérature, nos résultats dépassent souvent l'état-de-l'art, et ils en sont proches dans tous les cas. Nos solutions restent toutefois plus simples que les meilleurs modèles de la littérature.

ABSTRACT

Modeling a label global context for sequence tagging in recurrent neural networks

During the last few years Recurrent Neural Networks (RNN) have reached state-of-the-art performances on most sequence modeling problems. In particular the *sequence to sequence* model and the neural CRF have proved very effective on this class of problems. In this paper we propose an alternative RNN for sequence labelling, based on label embeddings and memory networks, which makes possible to take arbitrary long contexts into account. Our results are better than those of state-of-the-art models in most cases, and close to them in all cases. Moreover, our solution is simpler than the best models in the literature.

MOTS-CLÉS : Réseaux neuronaux récurrents, contexte global, Étiquetage de séquences.

KEYWORDS: Recurrent Neural Networks, global context, Sequence Labeling.

1 Introduction

L'étiquetage de séquences est un problème très important du TAL. En effet, beaucoup de problèmes de TAL peuvent être reformulés comme des problèmes d'étiquetage de séquences. Cette reformulation peut être intégrale dans certains cas, comme pour l'étiquetage en parties du discours (*POS tagging*), la segmentation syntaxique, la reconnaissance d'entités nommées (Collobert *et al.*, 2011) ou la compréhension automatique de la parole dans les systèmes de dialogue humain-machine (De Mori *et al.*, 2008). Dans d'autre cas, elle ne concerne que la première de plusieurs étapes, comme pour l'analyse syntaxique en constituants, qui peut être décomposée en étiquetage en parties du discours et en analyse des composants (Collins, 1997) ; la détection de chaînes de coréférences (Soon *et al.*, 2001; Ng & Cardie, 2002), décomposée en détection de mentions et détection des paires de mentions coréférentes ; mais aussi la détection d'entités nommées étendues (Grouin *et al.*, 2011), décomposée en détection des composants simples d'entités nommées,

combinés ensuite en entités nommées structurées plus complexes (Dinarelli & Rosset, 2011, 2012).

Dans cet article, nous nous intéressons aux modèles neuronaux pour l'étiquetage de séquences tel que la compréhension automatique de la parole, l'annotation en parties du discours et la détection d'entités nommées. Des modèles très efficaces existent pour ce type de problèmes, notamment le modèle *sequence to sequence* (Sutskever *et al.*, 2014) et toute la famille de modèles employant une couche de sortie de type CRF neuronal au dessus d'une ou plusieurs couches cachées récurrentes telles que *LSTM* ou *GRU* (Hochreiter & Schmidhuber, 1997; Cho *et al.*, 2014; Lample *et al.*, 2016; Ma & Hovy, 2016; Vukotic *et al.*, 2016; Chiu & Nichols, 2015; Huang *et al.*, 2015). Ces dernières solutions ont été motivées par la nécessité de remplacer dans les réseaux neuronaux la fonction locale *softmax*, moins adaptée aux problèmes sur les séquences, par une fonction de décision globale. Pour ces tâches, les CRF *classiques* (Lafferty *et al.*, 2001; Lavergne *et al.*, 2010) avaient déjà largement montré l'intérêt de la prise en compte des dépendances entre les unités de sortie.

Nous proposons une architecture neuronale alternative aux deux mentionnées ci-dessus. Cette architecture utilise une couche cachée de type *GRU* comme mémoire interne du réseau pour prendre en compte un contexte arbitrairement long, et ce aussi bien pour les unités d'entrée (les mots), que pour unités de sortie (les étiquettes). Pour une meilleure prise en compte de leurs dépendances, nous utilisons des plongements d'étiquettes en nous inspirant de la solution décrite dans (Dupont *et al.*, 2017; Dinarelli *et al.*, 2017). Ces deux choix architecturaux permettent de modéliser efficacement à la fois l'espace sémantique des étiquettes et leur contexte global, de la même façon que les modèles *sequence to sequence* (Sutskever *et al.*, 2014) ou le *LSTM+CRF* (Lample *et al.*, 2016).

Nous comparons notre solution avec l'état-de-l'art, notamment avec les modèles décrits dans (Dinarelli *et al.*, 2017) et (Lample *et al.*, 2016). À notre connaissance, bien que le modèle *sequence to sequence* a été utilisé pour des tâches d'étiquetage de séquences, il s'agissait de tâches différentes par rapport à celles auxquelles nous nous intéressons dans cet article. Pour avoir une comparaison équitable, nous nous comparons sur les mêmes tâches que (Dinarelli *et al.*, 2017) : deux tâches de compréhension automatique de la parole qui peuvent être modélisés comme des étiquetages de séquences : ATIS (Dahl *et al.*, 1994) et MEDIA (Bonneau-Maynard *et al.*, 2006).

Nos résultats dépassent dans la plupart des cas l'état-de-l'art, et ils en sont proches dans tous les cas. De plus notre solution est plus simple que les modèles *sequence to sequence* et *LSTM+CRF*, et, grâce à l'utilisation de technologies récentes [1], elle peut passer à l'échelle sur des quantités de données plus importantes que celles utilisées dans cet article.

2 Réseaux neuronaux alternatifs pour l'étiquetage de séquences

Les modèles neuronaux alternatifs aux modèles *sequence to sequence* et *LSTM+CRF* proposés dans cet article s'inspirent des modèles décrits dans (Dinarelli *et al.*, 2017). La similarité consiste dans l'utilisation de plongements d'étiquettes pour la représentation des unités de sortie.

Les modèles décrits dans (Dinarelli *et al.*, 2017) sont cependant assez simples. Que ce soit au niveau des mots ou au niveau des étiquettes, la prise en compte d'un contexte utilise une fenêtre de taille fixe. Ce choix limite la modélisation d'un contexte à la distance de la taille de la fenêtre choisie. Aussi, les modèles présentés dans (Dinarelli *et al.*, 2017) utilisent une couche cachée simple de type *ReLU* (Bengio, 2012), et n'utilisent pas l'algorithme *Back-Propagation Through Time* (BPTT) (Werbos, 1990), ce qui limite également la modélisation d'un contexte arbitrairement long.

1. Pytorch (Paszke *et al.*, 2017)

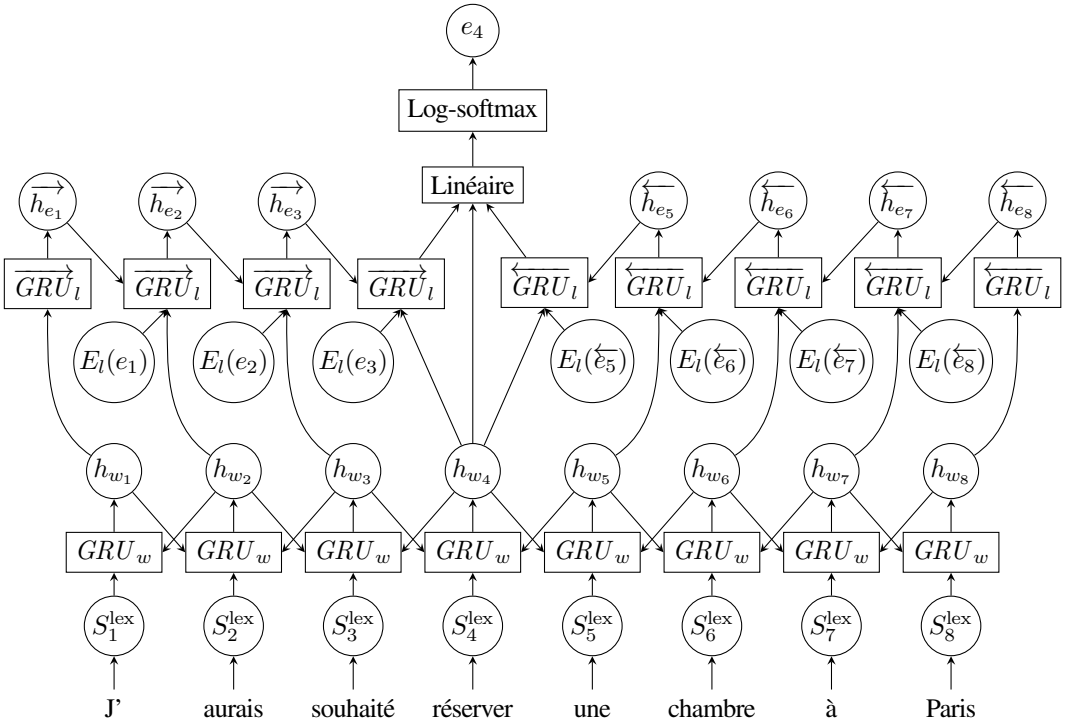

FIGURE 1 – Structure (simplifiée) du réseau

Les modèles que nous proposons dans cet article élimine ces restrictions en utilisant des couches cachées de type *GRU*. Celles-ci sont une évolution des couches *LSTM* qui donnent en général des meilleurs résultats (Cho *et al.*, 2014; Vukotic *et al.*, 2016), et montrent une meilleure capacité à mémoriser l'information contextuelle.

Dans la suite, $GRU(x_t,h_{t-1})$ désigne une couche cachée *GRU* bidirectionnelle, et $\overrightarrow{GRU}(x_t,h_{t-1})$ et $\overleftarrow{GRU}(x_t,h_{t-1})$ indiquent respectivement une couche cachée *forward* ou *backward*. Les sorties de ces couches seront notées respectivement $\mathbf{h_t}$, $\overrightarrow{h_t}$ et $\overleftarrow{h_t}$, une lettre en exposant indiquant le type d'entrée à partir de laquelle la couche cachée a été calculée. Par exemple $\overrightarrow{h_t^l}$ désigne la sortie de la couche cachée *forward* sur les étiquettes.

Les modèles présentés dans cet article utilisent toujours comme entrée x_t les mots, leurs caractères et les étiquettes. Nous utilisons pour les mots des plongements notés E_w, et pour les étiquettes des plongements notés E_l.

2.1 Représentation des mots au niveau des caractères

La représentation des mots au niveau des caractères est construite de la même façon que dans (Ma & Hovy, 2016), en utilisant une couche de type *GRU* au lieu d'une couche *LSTM*. Dans ce modèle, les caractères d'un mot $w = ch_1,...ch_{|w|}$ sont d'abord convertis en plongements. La couche GRU^{ch} est ensuite appliquée

à la séquence de plongements et son état final est retenu comme représentation du mot au niveau des caractères. Formellement :

$$\mathbf{W_{ch}} = (\mathbf{E_{ch}}(ch_1)...\mathbf{E_{ch}}(ch_{|w|})) \quad (1)$$

$$\mathbf{h^{ch}_{|w|}} = GRU_{ch}(\mathbf{W_{ch}}, h_0) \quad (2)$$

Où $\mathbf{E_{ch}}$ est la matrice des plongements de caractères, $\mathbf{W_{ch}}$ la séquence de plongements pour w, h_0 la valeur initiale de la couche cachée, et $\mathbf{h^{ch}_{|w|}}$ est sa valeur finale, utilisée comme représentation du mot w.

2.2 Représentation des mots

Les mots sont convertis en plongements et ensuite traités par une couche cachée GRU_w. Avec le même formalisme que pour les caractères, la séquence de mots $s = w_1...w_N$ est convertie en plongements $\mathbf{E_w}(w_i)$. On note $s_i = w_1...w_i$ la sous-séquence de s jusqu'au mot w_i. Pour augmenter la représentation des mots, les représentations au niveau des caractères $\mathbf{h^{ch}_{|w_i|}}$ sont concaténées avec les plongements de mots et le résultat est donné en entrée à la couche GRU_w. La représentation en contexte du mot w_i est calculée comme suit :

$$\mathbf{S^e} = (\mathbf{E_w}(w_1),...,\mathbf{E_w}(w_N)) \quad (3)$$

$$\mathbf{S^{lex}} = ([\mathbf{E_w}(w_1), \mathbf{h^{ch}_{|w_1|}}]...[\mathbf{E_w}(w_N), \mathbf{h^{ch}_{|w_N|}}]) \quad (4)$$

$$\mathbf{h_{w_i}} = GRU_w(\mathbf{S^{lex}_i}, h_{i-1}), \forall i \in [1...N] \quad (5)$$

Où $\mathbf{S^e}$ est la séquence de plongements construite à partir de la séquence s, $\mathbf{S^{lex}}$ est la séquence obtenue en concaténant plongements de mots et représentations au niveau des caractères, qui constitue l'information lexicale et $[\,]$ indique la concaténation de vecteurs. On note aussi $\mathbf{S^{lex}_i}$ la sous-séquence de $\mathbf{S^{lex}}$ jusqu'à la position i.

2.3 Représentation des étiquettes

Afin d'obtenir une représentation au niveau des étiquettes qui encode également un long contexte, nous utilisons une couche cachée *GRU* sur les plongements d'étiquettes. Nous faisons d'abord une passe *backward* pour calculer la représentation du contexte droit d'une étiquette à prédire donnée. On note $\overleftarrow{e_i}$ ces étiquettes prédites en utilisant uniquement le contexte droit. Formellement :

$$\overleftarrow{h_{e_i}} = \overleftarrow{GRU}_l(\mathbf{E_l}(\overleftarrow{e_{i+1}}), \overleftarrow{h_{e_{i+1}}}) \quad (6)$$

La sortie $\overleftarrow{h_{e_i}}$ est utilisée comme représentation du contexte droit pour prédire l'étiquette $\overleftarrow{e_i}$. Cette même représentation est utilisée aussi dans la phase *forward*, pendant laquelle le modèle dispose à la fois du contexte gauche et du contexte droit pour prédire l'étiquette finale. Le calcul du contexte gauche s'effectue de façon similaire avec une couche $\overrightarrow{GRU}_l$.

La couche $\overleftarrow{GRU}_l$ (et donc aussi la couche $\overrightarrow{GRU}_l$) utilise explicitement un contexte d'une seule étiquette. Grâce au fonctionnement de la couche cachée GRU, l'état de la couche cachée $\overleftarrow{h_{e_{i+1}}}$ (et $\overrightarrow{h_{e_{i-1}}}$) encode implicitement toutes les étiquettes précédentes.

Nous considérons que le contexte lexical est utile non seulement pour désambiguïser le mot courant à étiqueter, mais également pour désambiguïser le contexte d'étiquettes. En effet, les étiquettes constituent une information sémantique abstraite, dont il est raisonnable de penser qu'elle ne suffit pas pour discriminer les traits extraits pour obtenir la représentation des contextes d'étiquettes $\overleftarrow{h_{l_i}}$ et $\overrightarrow{h_{l_i}}$.

Nous ajoutons alors à l'entrée des couches $\overleftarrow{GRU}_l$ et $\overrightarrow{GRU}_l$, l'information lexicale $\mathbf{h_{w_i}}$ décrite plus haut. Avec cette modification, le calcul du contexte droit au niveau des étiquettes devient :

$$\overleftarrow{h_{e_i}} = \overleftarrow{GRU}_l([\mathbf{h_{w_i}}, \mathbf{E_l}(\overleftarrow{e_{i+1}})], \overleftarrow{h_{e_{i+1}}}) \tag{7}$$

Le calcul du contexte gauche s'effectue de façon similaire.

Notre choix d'utiliser l'information $\mathbf{h_{w_i}}$ dans les couches $\overleftarrow{GRU}_l$ et $\overrightarrow{GRU}_l$ est également motivé par la théorie des systèmes complexes, comme proposé dans (Wang, 2017). (Arthur, 1993) caractérise qualitativement l'évolution du fonctionnement d'un système complexe avec trois types d'adaptations différentes, notamment l'*agrégation* et la *spécialisation*.

Dans ce cadre, nous décrivons l'évolution de nos modèles en terme de *spécialisation*. Les cas les plus clairs sont ceux des portes (*gates*) des couches *LSTM* et *GRU*. En effet, les portes $\mathbf{z}$ et $\mathbf{r}$ de la couche *GRU* (cf. (Cho *et al.*, 2014)) sont définies exactement de la même façon, avec le même nombre de paramètres, et utilisent exactement les mêmes informations d'entrée. Pendant l'évolution du système (l'apprentissage), $\mathbf{r}$ s'adapte pour devenir la *reset gate*, qui permet d'*oublier* l'information passée quand celle-ci n'est pas pertinente, alors que $\mathbf{z}$ devient l'équivalent de l'*input gate* des *LSTM*, qui permet de contrôler l'information en entrée qui va être utilisée pour affecter la prédiction du modèle.

Du point de vue de la *spécialisation*, les couches $\overleftarrow{GRU}_l$ et $\overrightarrow{GRU}_l$ s'adaptent pour fonctionner comme une porte qui permet de filtrer au niveau des étiquettes l'information qui n'est pas pertinente pour la prédiction du modèle. De la même façon que les portes qui ont besoin à la fois de l'information d'entrée et de la valeur de la couche cachée précédente pour un fonctionnement optimal, la couche $\overleftarrow{GRU}_l$ utilise à la fois l'information lexicale et les étiquettes pour mieux discriminer l'information sémantique des étiquettes. Nous montrerons dans l'évaluation l'efficacité de ce choix architectural.

2.4 Apprentissage

Nous apprenons nos modèles en maximisant la log-vraisemblance avec les données :

$$LL(\Theta|D) = \sum_{d=1}^{|D|} \sum_{i=1}^{N_d} \log(P_\Theta(e_i|w_i, H_i) + \frac{\lambda}{2}|\Theta|^2) \tag{8}$$

Où les deux sommes s'appliquent aux données d'apprentissage et à chaque séquences des données. Les log-probabilités $\log(P_\Theta(e_i|w_i, H_i))$ sont calculées avec le *log-softmax* comme couche de sortie du réseau. La valeur H_i représente l'information contextuelle de nos modèles, c'est-à-dire l'information lexicale $\mathbf{h_{w_i}}$ et les

MEDIA			ATIS		
Mots	**Classes**	**Étiquettes**	**Mots**	**Classes**	**Étiquettes**
Oui	-	Answer-B	i'd	-	O
l'	-	BDObject-B	like	-	O
hotel	-	BDObject-I	to	-	O
le	-	Object-B	fly	-	O
prix	-	Object-I	Delta	airline	airline-name
à	-	Comp.-payment-B	between	-	O
moins	relative	Comp.-payment-I	Boston	city	fromloc.city
cinquante	tens	Paym.-amount-B	and	-	O
cinq	units	Paym.-amount-I	Chicago	city	toloc.city
euros	currency	Paym.-currency-B			

TABLE 1 – Un exemple d'annotation pris du corpus MEDIA (gauche) et ATIS (droite).

contextes *forward* et *backward* au niveau des étiquettes $\overrightarrow{h_{e_i}}$ et $\overleftarrow{h_{e_i}}$. Étant donnée la taille relativement petite des données sur lesquelles nous nous évaluons et la relative complexité des modèles, nous utilisons un terme de régularisation L_2, dont λ est le coefficient. La fonction de coût est minimisée par descente de gradient stochastique, le gradient étant estimé avec l'algorithme *Back-propagation Through Time* (Werbos, 1990).

3 Évaluation

3.1 Données utilisées et réglages

Nous évaluons nos modèles sur les deux tâches de compréhension de la parole **ATIS** (*Air Travel Information System*) (Dahl *et al.*, 1994) et **MEDIA** (Bonneau-Maynard *et al.*, 2006).

Ces deux tâches sont celles employées pour l'évaluation des modèles auxquels nous nous comparons (Dinarelli *et al.*, 2017). Nous renvoyons les lecteurs à ces travaux pour plus de détails sur les corpus. Un exemple comparatif d'annotation pris des deux corpus est montré dans le tableau 1.

Nous utilisons globalement les mêmes réglages utilisés dans (Dinarelli *et al.*, 2017), sauf pour certains d'entre eux que nous avons re-optimisés sur les données de développement : les plongements des étiquettes ont une taille de 150, les couches cachées ont une taille de 300, le *dropout* sur tous les plongements est de 0,5. Comme dans (Dinarelli *et al.*, 2017) également, nous utilisons les étiquettes *gold* pendant l'apprentissage du modèle.

3.2 Résultats

Les résultats montrés sont des moyennes sur 10 expériences. Pour avoir une comparaison équitable, nos réglages sont les mêmes que ceux utilisés dans (Dinarelli *et al.*, 2017). En revanche nous n'utilisons pas les classes de mots disponibles pour les deux tâches (cf. tableau 1) afin de nous placer dans un contexte plus réaliste. Pour réduire la quantité de mémoire utilisée, nous limitons la taille du contexte utilisé avec un hyper-paramètre dont la valeur est 10 par défaut. Nous avons aussi téléchargé le logiciel décrit dans (Dinarelli *et al.*, 2017) [2] et nous avons effectué des expériences par nous mêmes, sans utiliser les classes de mots comme trait des modèles.

Les résultats sont donnés en termes de précision, qui constitue le critère de choix du modèle en phase

2. Décrit à la page http ://www.marcodinarelli.it/software.php et disponible sous requête

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**

TABLE 2 – Comparaison des résultats obtenus sur les données de développement de la tâche MEDIA sans (GRU+IRNN) et avec l'information lexicale (GRU+IRNN$_{le}$) en entrée des couches $\overleftarrow{GRU}_l$ et $\overrightarrow{GRU}_l$

d'apprentissage sur les données de développement, en plus de la mesure *F1* et du taux d'erreur sur les étiquettes (***C**oncept **E**rror **R**ate*). Puisque notre modèle constitue une amélioration du modèle LD-RNN décrit dans (Dinarelli *et al.*, 2017), amélioration due notamment à l'utilisation des couches cachées GRU, dans la suite de notre article il sera indiqué avec *GRU+LD-RNN*.

Afin de montrer la capacité de nos modèles à prendre en compte un contexte global, ainsi que leur capacité à discriminer l'information pertinente pour la décision à un instant donné, nous montrons les résultats de deux expériences visant à confirmer cette capacité.

Dans la première expérience nous comparons les résultats obtenus par nos modèles sans et avec l'utilisation de l'information lexicale au niveau des couches $\overleftarrow{GRU}_l$ et $\overrightarrow{GRU}_l$ (cf. section 2.3). Ces résultats sont montrés dans le tableau 2. Le modèle utilisant l'information lexicale est indiqué avec GRU+LD-RNN$_{le}$ (pour information **l**exicale et **é**tiquettes). Ce modèle est meilleur que le modèle n'employant pas l'information lexicale, ce qui confirme que cette information est très importante pour distinguer l'information sémantique pertinente à un instant donné.

Dans la seconde expérience nous testons la capacité de nos modèles à filtrer l'information sémantique non pertinente pour la décision du modèle. Pour faire cela, nous utilisons une taille de contexte en phase d'apprentissage plus grande : 15 au lieu de 10 dans les expériences précédentes. Il est important de noter que dans un contexte de compréhension de la parole, dans lequel les données sont des transcriptions de l'oral, allonger le contexte est assez risqué puisque un contexte plus long contient à la fois plus d'information et plus de bruit. Par ailleurs, les modèles de la littérature employant une fenêtre de taille fixe, ne vont jamais au delà de 3 *token* par rapport à la position courante, ce qui confirme la difficulté à extraire de l'information utile de contextes plus longs. Les résultats de la seconde expérience sont montrés dans le tableau 2. Encore une fois, notre hypothèse semble être confirmée, le modèle utilisant un contexte de taille 15 étant meilleur que le modèle utilisant la taille 10.Nous tenons d'ailleurs à souligner que les modèles employant des couches cachées LSTM ou GRU ont tendance à sur-apprendre les données. Par manque de temps nous n'avons pas ré-optimisé les hyper-paramètres quand nous utilisons un contexte de taille 15 en phase d'apprentissage. Une optimisation plus fine pourrait conduire à des meilleurs résultats. Au delà de ces considérations, les résultats du modèle *GRU+LD-RNN$_{le}$ seg-len 15* sont suffisamment meilleurs pour pouvoir confirmer notre hypothèse, d'autant plus qu'ils sont compétitifs par rapport aux résultats des meilleurs modèles de la littérature, comme nous le montrons dans la suite.

Notre hypothèse concernant la *spécialisation* dans l'évolution de notre modèle semble confirmée (cf. section 2.3). Le fait que le modèle GRU+LD-RNN$_{le}$ obtienne des meilleurs résultats que le modèle GRU+LD-RNN simple, est déjà une preuve. En effet si le modèle GRU+LD-RNN$_{le}$ donne plus d'importance à l'information lexicale qu'à l'information provenant des étiquettes au niveau des couches $\overleftarrow{GRU}_l$ et $\overrightarrow{GRU}_l$, les meilleurs résultats n'auraient pas une explication claire, puisque les deux modèles GRU+LD-RNN$_{le}$ et GRU+LD-RNN (cf. tableau 2) utilisent tous les deux l'information lexicale séparément (indiquée

Modèle	Précision	F1	CER
MEDIA DEV			
LD-RNN$_{deep}$	89.26	85.79	10.72
GRU+LD-RNN$_{le}$	89.42	86.09	10.58
GRU+LD-RNN$_{le}$ seg-len 15	**89.97**	**86.57**	**10.42**
MEDIA TEST			
LD-RNN$_{deep}$	89.51	87.31	**10.02**
GRU+LD-RNN$_{le}$	89.48	87.36	10.28
GRU+LD-RNN$_{le}$ seg-len 15	**89.57**	**87.50**	10.26

TABLE 3 – Comparaison des résultats obtenus sur les données de développement et de test de la tâche MEDIA entre le système LD-RNN$_{deep}$ testé par nous même , et notre système GRU+LD-RNN$_{le}$ en utilisant un contexte de longueur 15.

Modèle	Précision	F1	CER
MEDIA TEST			
BiGRU+CRF (Dinarelli *et al.*, 2017)	–	86.69	10.13
LD-RNN$_{deep}$ (Dinarelli *et al.*, 2017)	–	87.36	**9.8**
LD-RNN$_{deep}$	89.51	87.31	10.02
GRU+LD-RNN$_{le}$ seg-len 15	**89.57**	**87.50**	10.26

TABLE 4 – Comparaison entre les résultats obtenus sur la tâche MEDIA avec notre meilleur système, GRU+LD-RNN$_{le}$ en utilisant un contexte de taille 15, et les meilleurs résultats de la littérature

avec $\mathbf{h_{w_i}}$ dans l'équation 5). Puisque l'information des étiquettes seules est déjà prise en compte par le modèle GRU+LD-RNN, nous pouvons déduire que le modèle GRU+LD-RNN$_{le}$ est capable d'extraire une représentation sémantique plus adaptée au contexte, et ce même quand nous utilisons un contexte plus long.

Dans une autre série d'expériences nous avons comparé notre modèle avec celui publié dans (Dinarelli *et al.*, 2017). Nous avons obtenu le logiciel associé à l'article [3] et nous avons effectué des expériences sur les mêmes données (MEDIA) dans les mêmes conditions. Nous avons utilisé la variante profonde LD-RNN$_{deep}$ décrite dans l'article, qui donne les meilleurs résultats. Les résultats de ces expériences sont montrés dans le tableau 3. Comme nous pouvons le constater, sur les données de développement (MEDIA DEV) notre modèle est meilleur que celui publié dans (Dinarelli *et al.*, 2017) qui détient l'état-de-l'art sur les tâches ATIS et MEDIA. Ces résultats sont confirmés aussi sur les données de test (MEDIA TEST), même si les marges d'amélioration sont un peu réduites, et le modèle LD-RNN$_{deep}$ reste le meilleur en termes de taux d'erreur (CER).

Nous avons effectué une dernière série d'expérience sur les deux tâches considérées dans cet article avec notre meilleur modèle. Ceci afin de nous comparer avec les meilleurs modèles de la littérature, qui sont encore une fois ceux publiés dans (Dinarelli *et al.*, 2017). Notamment nous nous comparons aux modèles employant une couche CRF neuronale, qui constituent une solution alternative pour une prise de décision globale au niveau des étiquettes.

Les résultats de ces expériences sont montrés dans le tableau 4 pour la tâche MEDIA, et dans le tableau 5 pour ATIS. Concernant les résultats sur MEDIA, le seul résultat nouveau par rapport à ceux déjà discutés

3. Décrit à la page http ://www.marcodinarelli.it/software.php

Modèle	Précision	F1	CER
ATIS TEST			
MLP+CRF (Dinarelli *et al.*, 2017)	–	95.45	5.28
LD-RNN (Dinarelli *et al.*, 2017)	–	**95.74**	**4.91**
GRU+LD-RNN$_{le}$ seg-len 15	98.08	95.70	5.04

TABLE 5 – Comparaison entre les résultats obtenus sur la tâche ATIS avec notre meilleur système, GRU+LD-RNN$_{le}$ en utilisant un contexte de taille 15, et les meilleurs résultats de la littérature

est le meilleur taux d'erreur de 9,8 du modèle LD-RNN$_{deep}$ publié dans (Dinarelli *et al.*, 2017). Ces résultats sont obtenus cependant en utilisant les classes de mots disponibles pour les tâches. Notre modèle reste meilleur en termes de précision et mesure F1, constituant donc le nouvel état-de-l'art.

Concernant la tâche ATIS (tableau 5), notre modèle obtient des résultats légèrement inférieurs à ceux du modèle LD-RNN. Ceci confirme les résultats sur MEDIA en termes de taux d'erreur, alors qu'en termes de mesure F1, d'après les commentaires de (Vukotic *et al.*, 2015) justifiés par la taille réduite des données et par la simplicité de la tâche, la différence ne semble pas statistiquement significative. Un bon résultat dans le tableau 5 est que notre modèle reste plus compétitif que le modèle MLP+CRF employant une couche CRF neuronale. Ce résultat va renforcer celui que nous avons obtenu sur la tâche MEDIA, sur laquelle les solutions employant un contexte au niveau des étiquettes sous formes de représentation distributionnelle sont meilleures que les solutions employant la couche CRF neuronale.

4 Conclusion

En considérant tous les résultats discutés dans cet article d'un point de vue global, nous pouvons conclure que l'emploi de couches cachées GRU pour construire un contexte global au niveau des étiquettes, est la plupart du temps plus efficace que les autres solutions proposées pour les tâches étudiées ; en considérant d'un côté les solutions employant un contexte au niveau des étiquettes sous formes de représentation distributionnelle, c'est-à-direles modèles LD-RNN et GRU+LD-RNN, et d'un autre côté les modèles employant une couche CRF neuronale, nous pouvons affirmer que les premières sont plus efficaces, du moins sur les tâches utilisées dans cet article pour l'évaluation, et représentent donc des solutions alternatives intéressantes et prometteuses pour l'étiquetage de séquences dans un sens plus général.

5 Remerciements

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

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

Références

ARTHUR W. B. (1993). *On the Evolution of Complexity*. Working papers, Santa Fe Institute.

BENGIO Y. (2012). Practical recommendations for gradient-based training of deep architectures. *CoRR*, **abs/1206.5533**.

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.

CHIU J. P. C. & NICHOLS E. (2015). Named entity recognition with bidirectional lstm-cnns. *CoRR*, **abs/1511.08308**.

CHO K., VAN MERRIENBOER B., GÜLÇEHRE Ç., BOUGARES F., SCHWENK H. & BENGIO Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, **abs/1406.1078**.

COLLINS M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, 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**, 2493–2537.

DAHL D. A., BATES M., BROWN M., FISHER W., HUNICKE-SMITH K., PALLETT D., PAO C., RUDNICKY A. & SHRIBERG E. (1994). Expanding the scope of the atis task : The atis-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, p. 43–48, Stroudsburg, PA, USA : Association for Computational Linguistics.

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.

DINARELLI M. & ROSSET S. (2011). Models cascade for tree-structured named entity detection. In *Proceedings of International Joint Conference of Natural Language Processing (IJCNLP)*, Chiang Mai, Thailand.

DINARELLI M. & ROSSET S. (2012). 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., 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 the 18th International Conference on Computational Linguistics and Intelligent Text Processing*, Budapest, Hungary : Lecture Notes in Computer Science (Springer).

GROUIN C., ROSSET S., ZWEIGENBAUM P., FORT K., GALIBERT O. & QUINTARD L. (2011). Proposal for an extension or traditional named entities : From guidelines to evaluation, an overview. In *Proceedings of the Linguistic Annotation Workshop (LAW)*.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural Comput.*, **9**(8), 1735–1780.

HUANG Z., XU W. & YU K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv :1508.01991*.

LAFFERTY J., MCCALLUM A. & PEREIRA F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, p. 282–289, Williamstown, MA, USA.

LAMPLE G., BALLESTEROS M., SUBRAMANIAN S., KAWAKAMI K. & DYER C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv :1603.01360*.

LAVERGNE T., CAPPÉ O. & YVON F. (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 504–513 : Association for Computational Linguistics.

MA X. & HOVY E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

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

PASZKE A., GROSS S., CHINTALA S., CHANAN G., YANG E., DEVITO Z., LIN Z., DESMAISON A., ANTIGA L. & LERER A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

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.

SUTSKEVER I., VINYALS O. & LE Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, p. 3104–3112, Cambridge, MA, USA : MIT Press.

VUKOTIC V., RAYMOND C. & GRAVIER G. (2015). Is it time to switch to word embedding and recurrent neural networks for spoken language understanding ? In *InterSpeech*, Dresde, Germany.

VUKOTIC V., RAYMOND C. & GRAVIER G. (2016). A step beyond local observations with a dialog aware bidirectional GRU network for Spoken Language Understanding. In *Interspeech*, San Francisco, United States.

WANG C. (2017). Network of recurrent neural networks. *CoRR*, **abs/1710.03414**.

WERBOS P. (1990). Backpropagation through time : what does it do and how to do it. In *Proceedings of IEEE*, volume 78, p. 1550–1560.