Apprentissage discriminant des modèles continus de traduction

Quoc-Khanh Do[1, 2] Alexandre Allauzen[1, 2] François Yvon[1]
(1) LIMSI/CNRS, rue John von Neumann, Campus Universitaire Orsay 91 403 Orsay
(2) Université Paris Sud, 91 403 Orsay
prenom.nom@limsi.fr

Résumé. Alors que les réseaux neuronaux occupent une place de plus en plus importante dans le traitement automatique des langues, les méthodes d'apprentissage actuelles utilisent pour la plupart des critères qui sont décorrélés de l'application. Cet article propose un nouveau cadre d'apprentissage discriminant pour l'estimation des modèles continus de traduction. Ce cadre s'appuie sur la définition d'un critère d'optimisation permettant de prendre en compte d'une part la métrique utilisée pour l'évaluation de la traduction et d'autre part l'intégration de ces modèles au sein des systèmes de traduction automatique. De plus, cette méthode d'apprentissage est comparée aux critères existants d'estimation que sont le maximum de vraisemblance et l'estimation contrastive bruitée. Les expériences menées sur la tâches de traduction des séminaires TED Talks de l'anglais vers le français montrent la pertinence d'un cadre discriminant d'apprentissage, dont les performances restent toutefois très dépendantes du choix d'une stratégie d'initialisation idoine. Nous montrons qu'avec une initialisation judicieuse des gains significatifs en termes de scores BLEU peuvent être obtenus.

Abstract.

Discriminative Learning of Continuous Translation Models.
This paper proposes a new discriminative framework to train translation models based on neural network. This framework relies on the definition of a new objective function that allows us to introduce the evaluation metric in the learning process as well as to consider how the model interacts with the translation system. Moreover, this approach is compared with the state of the art estimation methods, such as the maximum likelihood criterion and the noise contrastive estimation. Experiments are carried out on the English to French translation task of TED Talks . The results show the efficiency of the proposed approach, whereas the initialization has a strong impact. We show that with a tailored initialization scheme significant improvements can be obtained in terms of BLEU scores.

Mots-clés : Modèle neuronal de traduction, traduction automatique par approche statistique, apprentissage discriminant.

Keywords: Neural network based translation model, statistical machine translation, discriminative learning.

1 Introduction

Les modèles neuronaux occupent aujourd'hui dans le traitement automatique des langues (TAL) une place importante car ils permettent grâce à leur caractère continu des avancées significatives dans de nombreux domaines applicatifs. Historiquement, les modèles de langue neuronaux ont été une des premières réalisations marquantes, avec des applications en reconnaissance automatique de la parole (RAP), depuis les travaux pionniers de (Nakamura *et al.*, 1990) jusqu'aux développements ultérieurs de (Bengio *et al.*, 2003; Schwenk, 2007; Mnih & Hinton, 2007; Le *et al.*, 2011; Mikolov *et al.*, 2011). Les modèles neuronaux ont été également appliqués à d'autres tâches complexes de modélisation linguistique, comme par exemple l'analyse syntaxique (Socher *et al.*, 2013), l'estimation de similarité sémantique (Huang *et al.*, 2012), les modèles d'alignement de mots (Yang *et al.*, 2013) ou encore en traduction automatique statistique (TAS) (Le *et al.*, 2012; Kalchbrenner & Blunsom, 2013; Devlin *et al.*, 2014; Cho *et al.*, 2014).

Une des caractéristiques importantes des modèles neuronaux pour le TAL est leur caractère continu. En effet les modèles d'apprentissage probabilistes usuels reposent sur une représentation discrète des unités linguistiques considérées (mots, syntagmes, etc.). Typiquement, pour un modèle de traduction à base de segments, l'occurrence d'un segment est considérée comme la réalisation d'une variable aléatoire discrète, dont l'espace de réalisation est l'ensemble des segments

observés dans les données d'apprentissage. Au sein de cet espace, il n'existe aucune relation entre les éléments permettant de modéliser une notion de similarité, par exemple sémantique ou syntaxique. Le caractère très inégal des distributions d'occurrences dans les textes implique que les modèles résultants sont souvent estimés à partir de petits nombres d'occurrences, qu'ils possèdent une faible capacité de généralisation et que la modélisation du contexte est très coûteuse et donc souvent à horizon très limité.

Par opposition, les modèles neuronaux (Bengio *et al.*, 2003) se caractérisent par une méthode d'estimation alternative qui se fonde sur une représentation *continue* des unités qu'ils modélisent et en particulier des mots [1]. Dans le cas par exemple d'un modèle de langue, chaque mot du vocabulaire est représenté comme un point dans un espace métrique. La probabilité n-gramme d'un mot est alors une fonction des représentations continues des mots qui composent son contexte. Ces représentations, ainsi que les paramètres de la fonction d'estimation, sont apprises conjointement par un réseau de neurones multi-couches ; une stratégie d'estimation qui permet que les mots partageant des similarités distributionnelles auront des représentations proches. Ainsi, ce type de modèle introduit la notion de similarité entre mots et son exploitation permet une meilleure exploitation des données textuelles. L'intégration de ce type de modèle a permis des améliorations systématiques et significatives des performances en RAP et en TAS (Schwenk, 2007; Le *et al.*, 2011, 2012). Les représentations continues peuvent de plus servir à de nombreuses tâches, comme par exemple l'étiquetage en parties du discours et en rôle sémantique (voir (Turian *et al.*, 2010; Collobert *et al.*, 2011) pour une vue d'ensemble).

De nombreux travaux récents proposent différents types de modèles de traduction. Une part importante est dédiée aux modèles n-grammes de traduction (Schwenk *et al.*, 2007; Le *et al.*, 2012; Devlin *et al.*, 2014). Néanmoins ces travaux ont en commun d'apprendre les modèles de manière à maximiser la vraisemblance mesurée sur les données d'apprentissage. Or ce critère est peu corrélé avec d'une part les métriques utilisées pour évaluer la traduction et d'autre part l'intégration de ces modèles au sein des système de TAS. De plus, cet estimateur oblige le modèle à être normalisé ce qui représente un coût computationnel prohibitif étant donnés les espaces de réalisation utilisés. Il est alors nécessaire d'avoir recours à des solutions permettant d'alléger ce coût, comme l'utilisation d'une couche de sortie structurée ou l'usage d'un critère alternatif permettant de contourner cette contrainte.

Les contributions de cet article sont d'une part de proposer un cadre discriminant pour l'apprentissage des modèles continus de traduction permettant d'orienter l'optimisation du modèle vers les difficultés du système de TAS et donc d'apprendre à discriminer les hypothèses considérées selon la métrique utilisée lors de l'évaluation. D'autre part, cette approche est comparée à deux méthodes d'estimation compétitives : le maximum de vraisemblance, et l'estimation contrastive bruitée. Les résultats expérimentaux montrent des gains significatifs en termes de scores BLEU, et donc l'intérêt d'un tel cadre d'apprentissage pour la TAS. Le reste de l'article est organisé de la manière suivante : la section 2 introduit les modèles continus de traduction qui seront utilisés dans ces travaux ; puis les différentes méthodes d'apprentissage étudiées sont décrites à la section 3, avec en particulier la méthode discriminante ; les résultats expérimentaux sont enfin présentés à la section 4.

2 Modèles neuronaux pour la traduction automatique

Cette section propose une vue d'ensemble des modèles continues de traduction tels que nous allons les utiliser dans ces travaux. Si ce type de modèle s'intègre naturellement dans l'approche n-gramme en traduction automatique, il peut également être utilisé avec les approches usuelles à base de segments (Do *et al.*, 2014b). Pour plus de détails sur ces modèles et leur intégration, le lecteur peut se reporter à (Le *et al.*, 2012; Schwenk, 2012).

2.1 Approche n-gramme en traduction automatique

L'approche n-gramme en traduction automatique est une variante de l'approche à base de segments (ou *phrase-based*) (Zens *et al.*, 2002; Koehn, 2010). Décrite dans (Casacuberta & Vidal, 2004) puis (Mariño *et al.*, 2006; Crego & Mariño, 2006), elle s'en distingue par une décomposition spécifique de la probabilité jointe d'une paire de phrases parallèles où l'on suppose que la phrase source a été réordonnée au préalable. Ainsi, notons $P(\mathbf{s}, \mathbf{t})$ cette probabilité jointe, où $\mathbf{s}$ est une phrase source de I mots $(s_1, ..., s_I)$ réordonnés, et $\mathbf{t}$ la phrase cible associée et composée de J mots cibles $(t_1, ..., t_J)$. Cette paire de phrases est décomposée en L unités bilingues appelées *tuples*, $(\mathbf{s}, \mathbf{t}) = (\mathrm{u}_1, ..., \mathrm{u}_L)$. Une illustration de cette décomposition est donnée Figure 1.

1. Les modèles neuronaux sont souvent qualifiés de modèles continus.

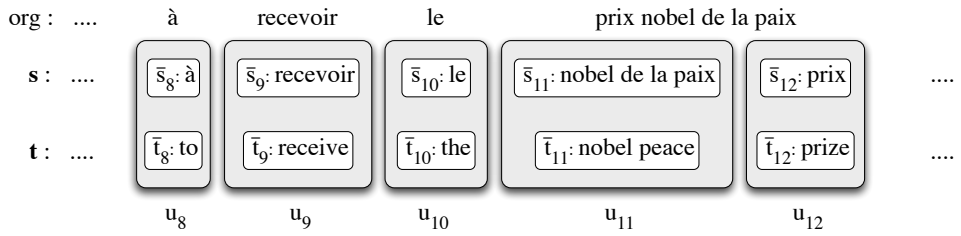


FIGURE 1: Extrait d'une paire de phrases parallèles segmentée. La phrase source originale (*org*) est indiquée au dessus de la phrase source réordonnée $\mathbf{s}$ et de la phrase cible $\mathbf{t}$. La paire de phrases $(\mathbf{s}, \mathbf{t})$ est décomposée en une séquence de L unités bilingues (*tuples*) $\mathrm{u}_1, ..., \mathrm{u}_L$. Chaque tuple u_i associe un segment source à un segment cible : $\overline{s}_i$ et $\overline{t}_i$.

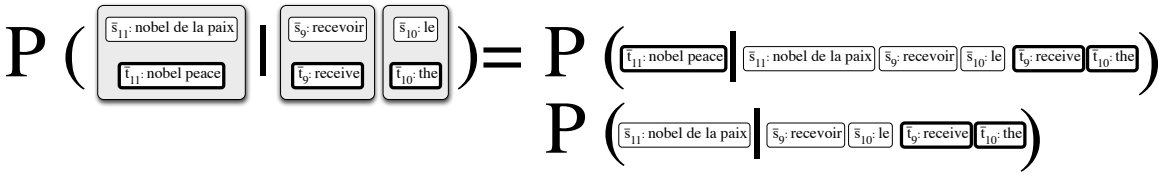


FIGURE 2: Exemple de décomposition en segment source et cible d'une paire de phrases parallèles sous l'hypothèse 3-gramme. Reprenant l'exemple de la figure 1, il s'agit de prédire le u_{11} connaissant u_9 et u_{10}.

Dans cette modélisation, les *tuples* sont les unités élémentaires de traduction [2], représentant une correspondance $\mathrm{u} = (\overline{s}, \overline{t})$ entre une séquence $\overline{s}$ de mots sources et une séquence de mots cibles $\overline{t}$. En utilisant l'hypothèse markovienne, la probabilité jointe peut être factorisée de la manière suivante :

$$P(\mathbf{s}, \mathbf{t}) = \prod_{i=1}^{L} P(\mathrm{u}_i | \mathrm{u}_{i-n+1}^{i-1}), \tag{1}$$

où u_{i-n+1}^{i-1} représente la séquence de tuples $\mathrm{u}_{i-n+1}, ..., \mathrm{u}_{i-1}$. Le modèle complet d'une paire de phrases parallèles contient donc les variables latentes précisant d'une part le réordonnancement de la phrase source, ainsi que la segmentation en unités bilingues. Ces variables latentes définissent la dérivation de la phrase source qui génère la phrase cible. Elles sont ignorées par la suite afin d'alléger les notations. Comme détaillé dans (Mariño *et al.*, 2006; Crego & Mariño, 2006), ces variables latentes sont inférées lors de la phase d'apprentissage à partir des données parallèles alignées automatiquement et ce en deux étapes : pour chaque paire de phrases parallèles, la phrase source est d'abord réordonnée de manière à suivre l'ordre des mots de la phrase cible, puis la segmentation en unités bilingues est effectuée.

Le modèle de traduction ainsi défini est un modèle de séquences utilisant l'hypothèse de n-gramme. La différence avec les modèles de langue monolingues est que les unités manipulées ne sont plus les mots mais les tuples. L'espace de réalisation considéré est alors bien plus grand qu'un inventaire monolingue de mots, alors que les données d'apprentissage disponibles se réduisent aux données parallèles. Ainsi, le caractère parcimonieux des données textuelles en général et des données parallèles en particulier rend difficile une estimation directe de ce type de modèle. Une solution est de décomposer les tuples en unités plus petites, comme, par exemple, en distinguant la partie source de la partie cible. L'équation (1) peut ainsi être décomposée de deux manières différentes :

$$\begin{aligned} & P(\mathrm{u}_i | \mathrm{u}_{i-n+1}^{i-1}) \\ & \quad = P(\overline{t}_i | \overline{s}_{i-n+1}^{i}, \overline{t}_{i-n+1}^{i-1}) P(\overline{s}_i | \overline{s}_{i-n+1}^{i-1}, \overline{t}_{i-n+1}^{i-1}) \\ & \quad = P(\overline{s}_i | \overline{t}_{i-n+1}^{i}, \overline{t}_{i-n+1}^{i-1}) P(\overline{t}_i | \overline{s}_{i-n+1}^{i-1}, \overline{t}_{i-n+1}^{i-1}) \end{aligned} \tag{2}$$

Considérons, par exemple, la première décomposition. Une illustration en est donnée à la figure 2. Désormais, deux espaces de réalisation sont impliqués, un par langue, qui recensent l'ensemble des segments. Il est encore possible de réduire ces espaces de réalisation en décomposant les segments en séquence de mots. Le modèle obtenu considère alors

2. Les tuples sont assimilables aux paires de segments ou bisegments (*phrase pairs*) utilisés dans l'approche plus classique à base de segments.

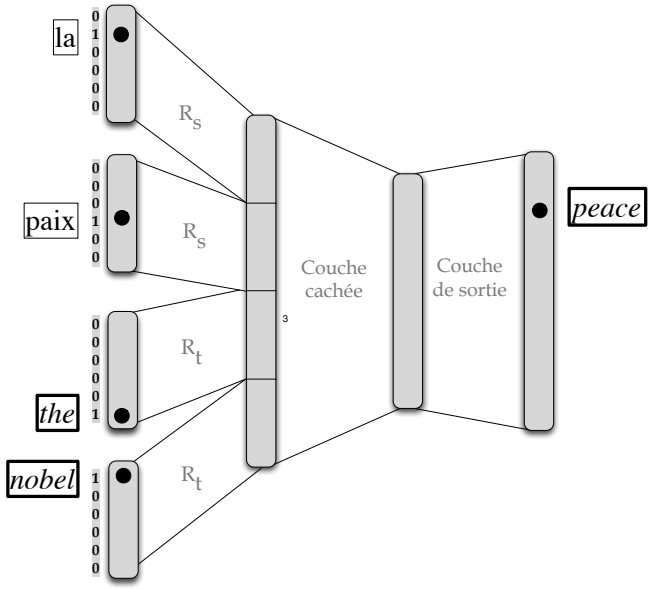


FIGURE 3: Architecture neuronale pour l'estimation des distributions n-grammes bilingues (ici $n = 3$). Cette figure illustre l'estimation de la distribution $P(\overline{t}_i|\overline{s}_{i-n+1}^{i}, \overline{t}_{i-n+1}^{i-1})$ relativement à l'exemple de la figure 1.

deux séquences de mots, l'une cible et l'autre source, synchronisées sur la segmentation en unités de traduction et dont la partie source a été réordonnée au préalable. Cela correspond à un modèle n-gramme bilingue de mots tel qu'il est initialement décrit dans (Le *et al.*, 2012) et étendu dans (Devlin *et al.*, 2014).

2.2 Architecture neuronale des modèles de traduction

L'estimation des distributions n-grammes peut être réalisée par des réseaux de neurones multi-couches, comme proposé dans (Bengio *et al.*, 2003; Schwenk, 2007) pour une application monolingue. Une architecture couramment utilisée est l'architecture *feed-forward*, illustrée sur la figure 3. Nous en résumons ici l'idée principale, les détails peuvent être trouvés dans (Le *et al.*, 2012) : les mots du contexte sont d'abord projetés dans un espace continu de représentation, chaque langue ayant sa matrice de projection (R_s et R_t respectivement pour les langues source et cible) ; leur concaténation permet d'obtenir une représentation continue du contexte bilingue ; puis une transformation non-linéaire est appliquée afin de prédire le mot cible grâce à la couche de sortie.

Avec ce type d'architecture, la taille du vocabulaire de sortie est la principale limitation [3]. Ainsi des solutions ont été proposées concernant spécifiquement la couche de sortie afin de réduire le coût d'inférence et d'apprentissage tout en permettant l'usage de vocabulaire de taille réaliste. La première consiste à structurer la couche de sortie comme proposé par (Mnih & Hinton, 2008). Dans cet article nous utilisons la structure SOUL décrite dans (Le *et al.*, 2011, 2013). Le modèle n-gramme peut être alors considéré comme un modèle neuronal de classe de mots. Une autre solution propose un cadre d'apprentissage pour des modèles non-normalisés, permettant de garder une couche de sortie de forme conventionnelle. Cette approche nommée estimation contrastive bruitée (ou NCE pour *noise contrastive estimation*) a été introduite par (Gutmann & Hyvärinen, 2010), puis appliquée aux modèles de langues par (Mnih & Teh, 2012), enfin intégrée à un système de traduction dans (Vaswani *et al.*, 2013). Cette approche sera détaillée à la section 3.1. Dans les deux cas, le modèle neuronal attribue un score positif à un mot w dans son contexte $\mathbf{c}$ noté $\mathbf{b}_{\boldsymbol{\theta}}(w, \mathbf{c})$, où $\boldsymbol{\theta}$ représente l'ensemble des paramètres du modèle à estimer. Concrètement, ce score positif est l'exponentiel de l'activation de la dernière couche linéaire du modèle, $\mathbf{b}_{\boldsymbol{\theta}}(w, \mathbf{c}) = \exp(\mathbf{a}_{\boldsymbol{\theta}}(w, \mathbf{c}))$. Les scores $\mathbf{b}_{\boldsymbol{\theta}}(.)$ peuvent ensuite être normalisés de manière efficace dans le cas du modèle SOUL, ou utilisés tels quels dans le cas d'un modèle NCE.

3. La majeure partie du coût computationnel se situe en effet au niveau de la couche de sortie, où il est nécessaire de normaliser la distribution en effectuant la somme sur l'ensemble du vocabulaire.

3 Méthodes d'apprentissage des modèles de traduction

Les modèles de traduction neuronaux décrits dans la section 2 sont habituellement appris en maximisant la log-vraisemblance, ou plus récemment en utilisant l'estimation contrastive bruitée. Or ces critères d'apprentissage n'ont qu'un lien lointain avec, d'une part, leur utilisation usuelle en traduction automatique et, d'autre part, avec les métriques d'évaluation. En effet, à cause du coût computationnel qu'ils impliquent, les modèles neuronaux sont le plus souvent utilisés en post-traitement d'un système de traduction conventionnel. Leur rôle est alors d'aider le système à trier un ensemble d'hypothèses en se basant sur une mesure automatique de qualité de la traduction, la plupart du temps le score BLEU (Papineni *et al.*, 2002). Cette étape est en général nommée N-*best reranking*, soit la réévaluation des N-meilleures hypothèses.

Dans cette section, nous commençons par décrire les deux critères d'apprentissage habituels des modèles de traduction neuronaux, puis nous formalisons (§ 3.2) un algorithme d'apprentissage discriminant visant à estimer directement les paramètres du modèle de traduction, de manière à optimiser l'étape de réévaluation des N-meilleures hypothèses. Cette méthode s'appuie sur la définition d'une fonction objectif que nous présentons dans un troisième temps (§ 3.3).

3.1 Maximum de vraisemblance et estimation contrastive bruitée

Traditionnellement, les modèles neuronaux de traduction sont entraînés de manière à maximiser la vraisemblance. En pratique, les données d'apprentissage sont présentées comme un ensemble de n-grammes $\mathcal{S}_n$, et la fonction objectif à minimiser[4] est la suivante :

$$\mathcal{L}_{cll}(\boldsymbol{\theta}, \mathcal{S}_n) = \sum_{(w,\mathbf{c})\in\mathcal{S}_n} -\log \mathbf{p}_{\boldsymbol{\theta}}(w|\mathbf{c}) + \mathcal{R}(\boldsymbol{\theta}), \tag{3}$$

où $\mathcal{R}(\boldsymbol{\theta})$ est le terme de régularisation L_2 défini par $\mathcal{R}(\boldsymbol{\theta}) = \gamma \times \frac{||\boldsymbol{\theta}||^2}{2}$ et γ est l'hyper-paramètre associé. Ce critère $\mathcal{L}_{cll}(\boldsymbol{\theta}, \mathcal{S}_n)$ correspond en fait à la somme négative des log-probabilités conditionnelles des n-grammes contenus dans les données d'apprentissage. Pour calculer cette fonction objectif, il est nécessaire de normaliser la sortie du réseau de neurones sur tous les mots du vocabulaire $\mathcal{V}$, selon :

$$\mathbf{p}_{\boldsymbol{\theta}}(w|\mathbf{c}) = \frac{\mathbf{b}_{\boldsymbol{\theta}}(w, \mathbf{c})}{\sum_{w'\in\mathcal{V}} \mathbf{b}_{\boldsymbol{\theta}}(w', \mathbf{c})},$$

où $\mathbf{b}_{\boldsymbol{\theta}}(w, \mathbf{c}) = \exp(\mathbf{a}_{\boldsymbol{\theta}}(w, \mathbf{c}))$ et $\mathbf{a}_{\boldsymbol{\theta}}(w, \mathbf{c})$ désigne l'activité du neurone de sortie associé au mot w. La minimisation de $\mathcal{L}_{cll}(\boldsymbol{\theta}, \mathcal{S}_n)$ se fait par descente de gradient stochastique. Néanmoins, le coût de la normalisation peut être prohibitif pour les tailles de vocabulaires typiquement utilisées en traduction automatique, qui contiennent des dizaines, voire des centaines de milliers d'entrées. Dans cet article, nous utilisons le modèle SOUL proposé par (Le *et al.*, 2011) qui, grâce à une couche de sortie structurée en arbre, permet de ramener le temps de calcul à des niveaux raisonnables.

Une approche différente permet de contourner le calcul induit par la normalisation : l'estimation contrastive bruitée ou *Noise Contrastive Estimation* (Gutmann & Hyvärinen, 2010). L'idée principale est de reformuler le problème comme une tâche de classification binaire entre, d'une part, les exemples positifs rencontrées dans les données d'apprentissage et, d'autre part, des exemples négatifs générés artificiellement selon une distribution de bruit $\mathbf{p}_N(.)$. Soit $\mathcal{X}^w$ la variable aléatoire binaire indiquant si le mot w est un exemple positif ou négatif. Nous faisons de plus l'hypothèse (justifiée ci-dessous) que les échantillons négatifs sont *a priori* K fois plus fréquents que les positifs, alors les probabilités à priori des deux évènements sont données par :

$$\mathbf{p}(\mathcal{X}^{w'} = 1) = \frac{1}{K+1}; \mathbf{p}(\mathcal{X}^{w'} = 0) = \frac{K}{K+1}$$

En supposant de plus que $\mathbf{p}_{\boldsymbol{\theta}}$ est une bonne approximation de la distribution empirique des exemples positifs, il est possible d'écrire :

$$\mathbf{p}(w'|\mathbf{c}, \mathcal{X}^{w'} = 1) = \mathbf{p}_{\boldsymbol{\theta}}(w'|\mathbf{c})$$
$$\mathbf{p}(w'|\mathbf{c}, \mathcal{X}^{w'} = 0) = \mathbf{p}_N(w'),$$

4. Dans la suite de cet article nous adoptons la convention habituelle en apprentissage automatique qui consiste à formuler l'apprentissage comme la minimisation d'une fonction objectif. Ainsi maximiser la vraisemblance est équivalent à minimiser le critère défini par l'équation (3).

puis de déduire, en appliquant le théorème de Bayes, les probabilités à postériori suivantes :

$$\begin{aligned}\mathbf{p}(\mathcal{X}^{w'}=1|w',\mathbf{c}) &= \frac{\mathbf{p}_{\boldsymbol{\theta}}(w'|\mathbf{c})}{\mathbf{p}_{\boldsymbol{\theta}}(w'|\mathbf{c})+K\mathbf{p}_N(w')}\\ \mathbf{p}(\mathcal{X}^{w'}=0|w',\mathbf{c}) &= \frac{K\mathbf{p}_N(w')}{\mathbf{p}_{\boldsymbol{\theta}}(w'|\mathbf{c})+K\mathbf{p}_N(w')}.\end{aligned} \tag{4}$$

La fonction objectif à minimiser devient alors l'espérance de $-\log(\mathbf{p}(\mathcal{X}^{w'})|w',\mathbf{c})$ sur l'ensemble d'exemples constitué d'un unique exemple positif (w), auxquels sont associés K exemples négatifs $\{w_1^*,\ldots,w_K^*\}$. La fonction objectif s'écrit alors de la manière suivante (en ré-intégrant tous les mots observés) :

$$\mathcal{L}_{nce}(\boldsymbol{\theta},\mathcal{S}_n) = \sum_{(w,\mathbf{c})\in\mathcal{S}_n}\Big[-\log\frac{\mathbf{p}_{\boldsymbol{\theta}}(w|\mathbf{c})}{\mathbf{p}_{\boldsymbol{\theta}}(w|\mathbf{c})+K\mathbf{p}_N(w)} - \sum_{i=1}^{K}\log\frac{K\mathbf{p}_N(w_i^*)}{\mathbf{p}_{\boldsymbol{\theta}}(w_i^*|\mathbf{c})+K\mathbf{p}_N(w_i^*)}\Big] + \mathcal{R}(\boldsymbol{\theta}), \tag{5}$$

où $(w,\mathbf{c})$ est un n-gramme issu des données d'apprentissage et $(w_i^*)_{i=1}^K$ l'ensemble des K exemples négatifs qui lui sont associés. Ces exemples négatifs sont tirés aléatoirement de la distribution de bruit. Dans (Gutmann & Hyvärinen, 2010; Mnih & Teh, 2012), les auteurs insistent sur l'importance du choix de cette distribution de bruit. Il semble en effet nécessaire qu'elle soit proche de la distribution empirique, tout en permettant un échantillonnage efficace. Ainsi, le choix le plus répandu est d'utiliser la distribution unigramme sur les données d'apprentissage. Dans notre cas, il s'agit d'une distribution unigramme sur les mots cibles.

Notons que, dans l'équation (4), le terme $\mathbf{p}_{\boldsymbol{\theta}}(w|\mathbf{c})$ apparait au numérateur et au dénominateur. En faisant l'approximation que $\mathbf{p}_{\boldsymbol{\theta}}$ et $\mathbf{p}_N$ ont des normalisations proches, il est possible de se débarrasser du terme de normalisation, et donc de remplacer (avantageusement) $\mathbf{p}_{\boldsymbol{\theta}}(w|\mathbf{c})$ par $\mathbf{b}_{\boldsymbol{\theta}}(w,\mathbf{c})$. De plus, il est possible de montrer que lorsque K tend vers l'infini, cette fonction objectif tend vers $\mathcal{L}_{cll}$. Ainsi l'estimation contrastive bruitée est une méthode d'optimisation formalisant le calcul approché de la constante de normalisation en échantillonnant K exemples négatifs au lieu d'effectuer la somme sur l'ensemble du vocabulaire.

3.2 Méthode discriminante d'apprentissage pour la traduction automatique

Malgré les progrès récents, l'inférence avec un réseau de neurones reste trop coûteuse pour que ce type de modèle puisse être intégré au décodage aussi facilement que les modèles de langue discrets utilisés dans les systèmes de traduction automatique [5]. L'usage est donc d'utiliser ces modèles lors d'une seconde étape de réévaluation des N-meilleures hypothèses.

Afin de définir ce cadre, supposons que pour chaque phrase source $\mathbf{s}$ à traduire, le décodeur génère une liste des N meilleures hypothèses $\{\mathbf{h}_1,\mathbf{h}_2,...,\mathbf{h}_N\}$. Chaque hypothèse $\mathbf{h}_i$ est constituée d'une phrase cible $\mathbf{t}_i$ et de la dérivation $\mathbf{a}_i$ qui l'a engendrée [6] Elle est évaluée par le système de traduction grâce à la fonction suivante :

$$F_{\boldsymbol{\lambda}}(\mathbf{s},\mathbf{h}) = \sum_{k=1}^{M}\lambda_k f_k(\mathbf{s},\mathbf{h}), \tag{6}$$

où M fonctions caractéristiques (f_k) sont pondérées par un jeu de poids $\boldsymbol{\lambda}$. Les fonctions caractéristiques utilisées dans cet article sont similaires à celles que l'on peut trouver dans les systèmes usuels à base de segments (voir (Crego *et al.*, 2011) pour plus de précisions).

L'introduction d'un modèle continu lors de l'étape de réévaluation des hypothèses se traduit par l'ajout à $F_{\boldsymbol{\lambda}}(.)$ d'une fonction caractéristique supplémentaire $f_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{h})$, qui varie selon le modèle utilisé :

$$f_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}) = \begin{cases}\log\mathbf{p}_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}) = \sum\limits_{(w,\mathbf{c})\in(\mathbf{s},\mathbf{h})}\log\mathbf{p}_{\boldsymbol{\theta}}(w|\mathbf{c}) & \text{pour le modèle SOUL,}\\ \sum\limits_{(w,\mathbf{c})\in(\mathbf{s},\mathbf{h})}\log\mathbf{b}_{\boldsymbol{\theta}}(w,\mathbf{c}) = \sum\limits_{(w,\mathbf{c})\in(\mathbf{s},\mathbf{h})}\mathbf{a}_{\boldsymbol{\theta}}(w,\mathbf{c}) & \text{pour le modèle NCE.}\end{cases} \tag{7}$$

5. Notons néanmoins les tentatives récentes d'intégration des modèles neuronaux de traduction dans le décodeur (Niehues & Waibel, 2012; Vaswani *et al.*, 2013; Devlin *et al.*, 2014).

6. $\mathbf{a}_i$ regroupe l'ensemble des variables latentes du processus de traduction. Dans le cas d'un système de traduction n-gramme, il s'agit du réordonnancement de la phrase source et du choix des unités bilingues (cf. § 2.1).

Algorithm 1 Procédure d'optimisation jointe de $\boldsymbol{\theta}$ et $\boldsymbol{\lambda}$

```
Init. de θ et λ
Pour chaque itération faire
    Pour M paquets faire                                   ▷ λ fixé
        Calcul du sous-gradient de L(θ) pour chaque phrase s du paquet
        Mise à jour de θ
    Fin Pour
    Mise à jour de λ en utilisant le dev.                  ▷ θ fixé
Fin Pour
```

Dans les deux cas, il est nécessaire de prendre la somme sur tous les n-grammes extraits de la dérivation considérée. Comme précédemment, $\boldsymbol{\theta}$ désigne le vecteur de paramètres définissant le modèle continu de traduction. Ainsi la fonction d'évaluation devient :

$$G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}) = F_{\boldsymbol{\lambda}}(\mathbf{s},\mathbf{h}) + \lambda_{K+1} f_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}) \tag{8}$$

Cette fonction dépend à la fois des paramètres $\boldsymbol{\theta}$ du modèle continu de traduction et des paramètres de mélange $\boldsymbol{\lambda}$ de la fonction d'évaluation. Ainsi, dans l'approche que nous proposons, l'optimisation nécessite *d'alterner l'estimation des poids de mélange $\boldsymbol{\lambda}$ et l'apprentissage des paramètres $\boldsymbol{\theta}$ du modèle continu* : la première étape utilise classiquement les données de développement alors que la deuxième utilise les données parallèles d'apprentissage.

Cette procédure d'optimisation est décrite par l'algorithme 1. Les données d'apprentissage sont découpées en paquets de 128 phrases successives. Chacun de ces paquets sert à la mise à jour de $\boldsymbol{\theta}$ à $\boldsymbol{\lambda}$ constant et ces derniers sont réestimés tous les P paquets. Notons que cet algorithme nécessite la définition d'une fonction objectif $\mathcal{L}(\boldsymbol{\theta})$ pour le modèle continu de traduction qui sera décrite à la section 3.3. Dans cet article, l'optimisation de $\boldsymbol{\lambda}$ utilise les outils standards, en l'occurrence l'algorithme *K-Best Mira* décrit dans (Cherry & Foster, 2012) et tel qu'il est implémenté dans MOSES [7].

3.3 Une fonction objectif discriminante

Le critère discriminant d'apprentissage proposé dans cet article s'inspire à la fois des méthodes à vaste marge et des approches de *ranking*. Comme expliqué précédemment, chaque hypothèse de traduction $\mathbf{h}_i$ engendrée par le système de traduction est évaluée selon l'équation (8). Mais sa qualité peut également être évaluée selon un critère de qualité de traduction, ici le score BLEU, ou plus précisément selon une approximation du score BLEU au niveau de la phrase, que l'on note $sBLEU(\mathbf{h}_i)$. Si $\mathbf{h}^*$ désigne l'hypothèse ayant le meilleur score, il est possible de définir un critère visant à maximiser la marge (Freund & Schapire, 1999; McDonald *et al.*, 2005; Watanabe *et al.*, 2007) de la manière suivante :

$$\mathcal{L}_{mm}(\boldsymbol{\theta},\mathbf{s}) = -G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}^*) + \max_{1\leq j\leq N} \left(G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_j) + \text{cost}_\alpha(\mathbf{h}_j)\right), \tag{9}$$

où $\text{cost}_\alpha(\mathbf{h}_j) = \alpha\big(sBLEU(\mathbf{h}^*) - sBLEU(\mathbf{h}_j)\big)$ représente la fonction de coût et le paramètre α pondère sa contribution. Lorsque $\alpha = 0$, nous retrouvons la fonction objectif du perceptron structuré (Collins, 2002). Si ce critère introduit une marge entre $\mathbf{h}^*$ est les autres hypothèses, il existe parmi les autres hypothèses des traductions qui pourraient être acceptables et qu'il conviendrait de considérer autrement qu'en les jugeant mauvaises. Ainsi, une alternative est de s'inspirer du classement par paire (ou *pairwise ranking*) comme le propose le système PRO de Hopkins & May (2011). Supposons que r_i désigne le rang de l'hypothèse $\mathbf{h}_i$ lorsque la liste des hypothèses est triée avec comme critère $sBLEU$, il est alors possible de définir la fonction objectif suivante :

$$\mathcal{L}_{pro}(\boldsymbol{\theta},\mathbf{s}) = \sum_{1\leq i,k\leq N} \mathbb{I}_{\{r_i+\delta\leq r_k, G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_i) < G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_k)\}} \left(-G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_i) + G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_k)\right). \tag{10}$$

Notons que cette fonction objectif implique un sous-ensemble de $N(N-1)/2$ paires d'hypothèses. En effet, une paire d'hypothèses n'est prise en compte que si les hypothèses qui la composent sont suffisamment éloignées en terme de rang : formellement la différence absolue des rangs doit excéder un seuil prédéfini δ.

Le critère que nous proposons est une combinaison des deux critères précédents. Ce choix s'appuie sur les résultats expérimentaux de (Do *et al.*, 2014a), qui a introduit ces critères dans le cadre de l'adaptation de modèle. Considérons que

7. http://www.statmt.org/moses/

pour une paire d'hypothèses $(\mathbf{h}_i, \mathbf{h}_k)$ telle que $r_i + \delta < r_k$, la différence de score $G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k)$ doit être au-delà d'une certaine marge. Comme précédemment, la marge s'exprime grâce à l'approximation du score BLEU au niveau de la phrase et donc via la fonction de coût cost_α. Nous pouvons alors définir le sous-ensembles des hypothèses *critiques* comme :

$$\mathcal{C}_\delta^\alpha = \{(i,k) : 1 \leq i,k \leq N, r_i + \delta \leq r_k, G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k) < \text{cost}_\alpha(\mathbf{h}_k) - \text{cost}_\alpha(\mathbf{h}_i)\}. \tag{11}$$

La fonction objectif que nous allons utiliser pour apprendre les modèles de traduction continus se définit de la manière suivante :

$$\mathcal{L}_{pro-mm}(\boldsymbol{\theta}, \mathbf{s}) = \sum_{(i,k)\in\mathcal{C}_\delta^\alpha} \text{cost}_\alpha(\mathbf{h}_k) - \text{cost}_\alpha(\mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) + G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k). \tag{12}$$

Cette fonction objectif ne requiert pas, tout comme le NCE, que le score du modèle neuronal $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h})$ inclus dans $G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i)$ soit normalisé. Il est donc possible d'apprendre un modèle de type SOUL selon ce critère mais également un modèle non-normalisé de type NCE. Remarquons enfin que si $\alpha = 0$, cette fonction se ramène à celle du classement par paire de l'equation (10).

4 Expériences

Afin de comparer les différentes méthodes d'apprentissage présentées à la section 3, une série d'expériences est menée sur la tâche de traduction automatique anglais vers français des séminaires TED Talks . Cette tâche fait partie de la campagne d'évaluation internationale sur la traduction de la parole organisée dans le cadre des ateliers IWSLT [8].

4.1 Cadre expérimental

La tâche consisdérée est la traduction des séminaires TED Talks (Federico *et al.*, 2012) dans leur version transcrite manuellement. Les données parallèles d'apprentissage servant à l'apprentissage des modèles neuronaux contiennent 107058 paires de phrases. Les données de développement et de test contiennent respectivement 934 et 1664 paires de phrases. En suivant (Le *et al.*, 2012), ces données sont échangées ; les coefficients $\boldsymbol{\lambda}$ sont estimés à partir des 1664 paires de phrases et l'autre corpus sert à l'évaluation. Le critère d'évaluation de la traduction est le score BLEU (Papineni *et al.*, 2002).

Le système de traduction utilise une implémentation libre de l'approche n-gramme [9] et ses modèles ont été appris à partir de vaste quantité de données bilingues et monolingues dans le cadre de la campagne d'évaluation WMT. Le système est décrit plus précisément dans l'article (Allauzen *et al.*, 2013).

Les modèles continus de traduction sont appris uniquement sur les données TED Talks . Chaque modèle, SOUL et NCE, est initialisé à partir de modèles n-grammes monolingues estimés respectivement sur la partie source et cible du corpus bilingue. Tous les modèles n-grammes continus sont des 10-grammes. Pour l'apprentissage discriminant, le système de traduction est d'abord utilisé pour générer une liste des 300 meilleures hypothèses pour chaque phrase source. Le seuil δ (cf. l'équation (11)) a été empiriquement fixé à 250 en fonction des scores BLEU sur les données de développement. Ces scores servent aussi de choisir la meilleure itération qui correspond au modèle qui est ensuite évalué sur les données de test.

Comme décrit à la section 2.1, il existe deux manières de décomposer la probabilité jointe d'une paire de phrases (voir l'équation (2)), et il est donc possible de définir 4 modèles continus de traduction. Par souci de clarté, seul le modèle $P(\overline{t}_i | \overline{s}_{i-n+1}^i, \overline{t}_{i-n+1}^{i-1})$ est utilisé par la suite. Néanmoins des tendances similaires ont été observées avec les autres modèles.

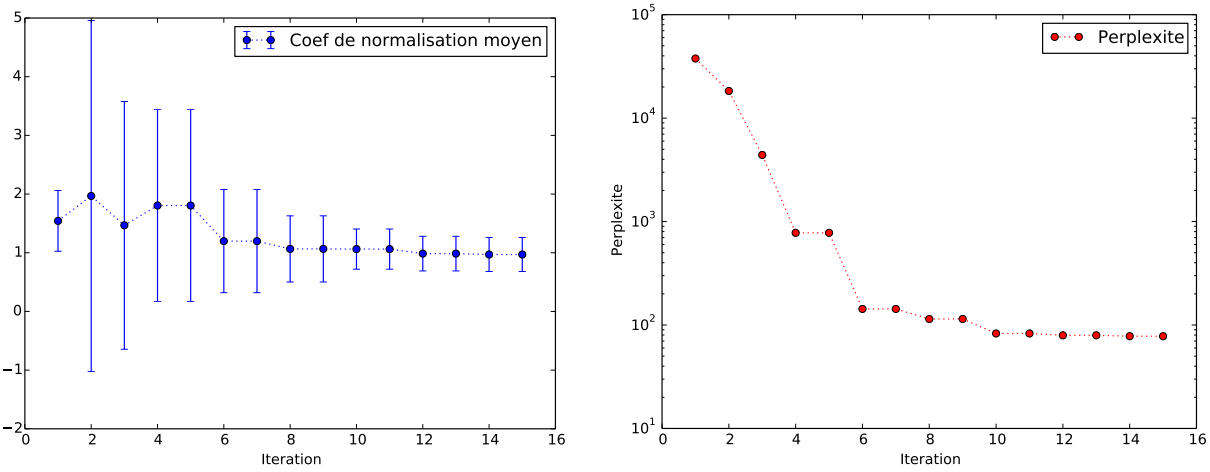
4.2 Résultats expérimentaux

Afin de comparer les différents critères décrits à la section 3, la première série d'expériences concerne les modèles non-normalisés selon qu'ils soient appris avec le NCE, le critère discriminant que nous proposons ou les deux. Dans tous les

8. International Workshop on Spoken Language Translation : `http://workshop2014.iwslt.org/`
9. `perso.limsi.fr/Individu/jmcrego/bincoder`

	dev	**test**
Système de traduction	33, 9	27, 6
Ajout d'un modèle continu standard		
NCE	35, 0	28, 8
Ajout d'un modèle continu discriminant		
initialisation aléatoire	34, 3	28, 4
initialisation monolingue NCE	35, 3	29, 0
NCE + discriminant	**35,4**	**29,7**
Oracle	46, 1	39, 0

TABLE 1: Comparaisons des résultats obtenus en terme de score BLEU avec différents modèles de traduction continus non normalisés.



(a) Évolution du coefficient de normalisation (moyenne et écart type)

(b) Évolution de la perplexité mesurée sur les données de validation.

FIGURE 4: Évolution de l'apprentissage d'un modèle NCE au cours du temps.

cas, le modèle neuronal est intégré via son score non-normalisé $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) = \sum_{(w,\mathbf{c})\in(\mathbf{s},\mathbf{h})} \mathbf{a}_{\boldsymbol{\theta}}(w, \mathbf{c})$. Le tableau 1 rassemble les résultats obtenus avec différents critère d'apprentissage et différentes configurations. Observons d'abord que le modèle continu de traduction appris avec le NCE comme critère permet d'obtenir une amélioration de $1, 2$ point BLEU et ce malgré l'utilisation d'une distribution de bruit relativement éloignée de la distribution empirique puisqu'il s'agit d'une distribution monolingue.

La seconde partie du tableau regroupe les différentes expériences utilisant le critère discriminant d'apprentissage et montre l'importance de l'initialisation pour ce critère d'apprentissage [10]. En partant d'une initialisation aléatoire, le modèle discriminant permet d'obtenir un gain significatif de 0,8 point BLEU mais qui est moindre que celui obtenu avec un modèle NCE. Par contre, si on utilise la même initialisation que celle du modèle NCE (voir section 4.1), le modèle discriminant apporte un gain supplémentaire par rapport au modèle NCE de 0,2 points BLEU. Enfin, le meilleur résultat est obtenu en combinant les deux critères : le modèle discriminant est appris avec en guise d'initialisation le modèle NCE. Avec cette configuration le modèle de traduction permet d'obtenir un gain de 2,1 points BLEU.

Afin de mieux comprendre comment fonctionne l'apprentissage NCE, la figure 4a représente l'évolution des coefficients de normalisation (moyenne et écart type) du modèle au cours des itérations . Un tel coefficient est calculé pour chaque contexte présent dans les données de validation [11]. Nous observons que la valeur moyenne, ainsi que l'écart type de ce coefficient, convergent rapidement. Néanmoins, l'écart type reste élevé, montrant ainsi que le modèle NCE n'est pas

10. Les modèles neuronaux, par leurs couches cachées donnent lieu à des fonctions objectifs non-convexes. Puisqu'une descente de gradient est utilisée lors de l'optimisation, le point de convergence dépend de l'initialisation.

11. Le coefficient de normalisation se calcul pour un contexte donné par $\sum_{w\in\mathcal{V}} \mathbf{b}_{\boldsymbol{\theta}}(w, \mathbf{c})$.

	dev	**test**
Système de traduction	$33,9$	$27,6$
Ajout d'un modèle continu standard		
SOUL	$35,1$	$28,9$
Ajout d'un modèle continu discriminant		
initialisation aléatoire	$33,8$	$27,7$
initialisation monolingue SOUL	$35,0$	$28,9$
SOUL + discriminant	$35,7$	$29,3$
Oracle	$46,1$	$39,0$

TABLE 2: Comparaison et utilisation des modèles SOUL dans le cadre discriminant d'apprentissage.

	SOUL	NCE	DISCRIM
Vitesse d'entraînement (mots/second)	1000	1000	
Nombre d'itérations	3	14	
Temps d'entraînement total, init incl. (heures)	9	9	15
Vitesse d'inférence (mots/second)	20000	25000	dépend du modèle

TABLE 3: Vitesse de traitement lors de l'apprentissage et de l'inférence, ainsi que le temps total d'apprentissage (comprenant la phase d'initialisation) des modèles décrits à la section 3. Si les vitesses d'entraînement des modèles SOUL et NCE sont équivalentes, l'inférence avec le modèle NCE est légèrement plus rapide. On note également que même si l'entraînement NCE demande plus d'itérations pour converger, son initialisation est bien plus simple par rapport à celle d'un modèle SOUL qui nécessite de construire une structure d'arbre pour tous les mots du vocabulaire.

normalisé. De plus, la figure 4b représente l'évolution de la perplexité [12] mesurée sur les mêmes données de validation. On constate également une convergence rapide et un comportement similaire à celui d'un modèle estimé selon le maximum de vraisemblance.

La seconde série d'expériences permet de comparer le critère usuel d'apprentissage qui est le maximum de vraisemblance et donc du modèle SOUL. Les résultats sont rassemblés dans le tableau 2. Remarquons tout d'abord qu'il n'y a qu'une différence de 0,1 points BLEU en faveur du modèle SOUL par rapport au modèle NCE. À ce stade, il est important de noter que les deux méthodes, SOUL et NCE, induisent des temps d'apprentissage équivalents pour les modèles de traduction. Ainsi, le caractère normalisé ne semble pas indispensable. Par contre, introduire le score normalisé du modèle SOUL semble moins favorable au cadre discriminant [13]. En effet, en partant d'une initialisation aléatoire, on notera qu'une très faible amélioration par rapport au résultat du système de traduction. De même, en partant d'une initialisation utilisant les modèles SOUL monolingues, l'apprentissage discriminant n'apporte rien par rapport au modèle SOUL bilingue. Le seul véritable gain est obtenu en combinant les deux critères, mais là encore, l'utilisation du NCE permet d'obtenir un meilleur résultat. Enfin, le tableau 3 rassemble les vitesses et temps de calcul liés à l'apprentissage et à l'inférence des différentes méthodes d'apprentissage décrites dans cet article.

5 Conclusions

Dans cet article nous avons proposé un cadre discriminant pour l'apprentissage des modèles neuronaux de traduction. Ce cadre s'appuie sur la définition d'un critère d'optimisation qui permet d'une part d'introduire la mesure servant à évaluer la traduction, et d'autre part de prendre en compte l'état courant du système de base pendant l'entraînement, contrairement aux autres méthodes existantes comme l'estimation au maximum de vraisemblance et l'estimation contrastive bruitée. Ces trois critères sont décrits puis comparés expérimentalement dans le cadre d'une tâche de traduction automatique de l'anglais vers le français des séminaires TED Talks . Les résultats montrent d'une part qu'il est possible d'apprendre un modèle continu de traduction de manière discriminante et d'autre part que le choix de l'initialisation revêt une grande importance. Nous proposons d'ailleurs à ce sujet des solutions efficaces permettant d'obtenir des gains significatifs en termes de scores BLEU. Notamment, la meilleure configuration consiste à enchaîner l'entraînement discriminant sur un

12. Le modèle original n'étant pas normalisé, il est nécessaire d'effectuer cette normalisation pour le calcul de la perplexité.
13. ici $f_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}) = \sum_{(w,\mathbf{c})\in(\mathbf{s},\mathbf{h})} \log \mathbf{p}_{\boldsymbol{\theta}}(w|\mathbf{c})$

modèle NCE à scores non-normalisés, ce qui dispense de normaliser les scores sur l'ensemble du vocabulaire. En guise de futurs travaux, il semble intéressant d'explorer ce cadre d'apprentissage pour des paires de langues peu dotées en données parallèles. En effet ce cadre semble permettre une meilleure exploitation des données d'apprentissage.

Références

ALLAUZEN A., PÉCHEUX N., DO Q. K., DINARELLI M., LAVERGNE T., MAX A., LE H.-S. & YVON F. (2013). LIMSI @ WMT13. In *Proceedings of the Workshkop on Statistical Machine Translation*, p. 62–69, Sofia, Bulgaria.

BENGIO Y., DUCHARME R., VINCENT P. & JAUVIN C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, **3**, 1137–1155.

CASACUBERTA F. & VIDAL E. (2004). Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, **30**(3), 205–225.

CHERRY C. & FOSTER G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, p. 427–436.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1724–1734, Doha, Qatar : Association for Computational Linguistics.

COLLINS M. (2002). Discriminative training methods for hidden Markov models : theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1–8.

COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, **12**, 2493–2537.

CREGO J. M. & MARIÑO J. B. (2006). Improving statistical MT by coupling reordering and decoding. *Machine Translation*, **20**(3), 199–215.

CREGO J. M., YVON F. & MARIÑO J. B. (2011). N-code : an open-source bilingual N-gram SMT toolkit. *Prague Bulletin of Mathematical Linguistics*, **96**, 49–58.

DEVLIN J., ZBIB R., HUANG Z., LAMAR T., SCHWARTZ R. & MAKHOUL J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1370–1380, Baltimore, Maryland.

DO Q. K., ALLAUZEN A. & YVON F. (2014a). Discriminative adaptation of continuous space translation models. In *International Workshop on Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, USA.

DO Q. K., HERRMANN T., NIEHUES J., ALLAUZEN A., YVON F. & WAIBEL A. (2014b). The KIT-LIMSI Translation System for WMT 2014. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, p. 84–89, Baltimore, Maryland, USA : Association for Computational Linguistics.

FEDERICO M., STÜKER S., BENTIVOGLI L., PAUL M., CETTOLO M., HERRMANN T., NIEHUES J. & MORETTI G. (2012). The IWSLT 2011 evaluation campaign on automatic talk translation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* : European Language Resources Association (ELRA).

FREUND Y. & SCHAPIRE R. E. (1999). Large margin classification using the perceptron algorithm. *Machine learning*, **37**(3), 277–296.

GUTMANN M. & HYVÄRINEN A. (2010). Noise-contrastive estimation : A new estimation principle for unnormalized statistical models. In Y. TEH & M. TITTERINGTON, Eds., *Proceedings of th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, p. 297–304.

HOPKINS M. & MAY J. (2011). Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 1352–1362, Edinburgh, Scotland, UK.

HUANG E., SOCHER R., MANNING C. & NG A. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 873–882, Jeju Island, Korea : Association for Computational Linguistics.

KALCHBRENNER N. & BLUNSOM P. (2013). Recurrent continuous translation models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1700–1709, Seattle, Washington, USA.

KOEHN P. (2010). *Statistical Machine Translation*. New York, NY, USA : Cambridge University Press, 1st edition.

LE H.-S., ALLAUZEN A. & YVON F. (2012). Continuous space translation models with neural networks. In *Proceedings of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, p. 39–48, Montréal, Canada.

LE H.-S., OPARIN I., ALLAUZEN A., GAUVAIN J.-L. & YVON F. (2011). Structured output layer neural network language model. In *Proceedings of ICASSP*, p. 5524–5527.

LE H.-S., OPARIN I., ALLAUZEN A., GAUVAIN J.-L. & YVON F. (2013). Structured output layer neural network language models for speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, **21**(1), 197–206.

MARIÑO J. B., BANCHS R. E., CREGO J. M., DE GISPERT A., LAMBERT P., FONOLLOSA J. A. & COSTA-JUSSÀ M. R. (2006). N-gram-based machine translation. *Computational Linguistics*, **32**(4), 527–549.

MCDONALD R., CRAMMER K. & PEREIRA F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 91–98.

MIKOLOV T., KOMBRINK S., BURGET L., CERNOCKÝ J. & KHUDANPUR S. (2011). Extensions of recurrent neural network language model. In *Proceedings of ICASSP*, p. 5528–5531.

MNIH A. & HINTON G. E. (2007). Three new graphical models for statistical language modelling. In *ICML*, p. 641–648.

MNIH A. & HINTON G. E. (2008). A scalable hierarchical distributed language model. In D. KOLLER, D. SCHUURMANS, Y. BENGIO & L. BOTTOU, Eds., *Advances in Neural Information Processing Systems 21*, volume 21, p. 1081–1088.

MNIH A. & TEH Y. W. (2012). A fast and simple algorithm for training neural probabilistic language models. In *ICML*.

NAKAMURA M., MARUYAMA K., KAWABATA T. & KIYOHIRO S. (1990). Neural network approach to word category prediction for english texts. In *Proceedings of the 13th conference on Computational linguistics (COLING)*, volume 3, p. 213–218.

NIEHUES J. & WAIBEL A. (2012). Continuous space language models using restricted Boltzmann machines. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, p. 164–170, Hong-Kong, China.

PAPINENI K., ROUKOS S., WARD T. & ZHU W. J. (2002). Bleu : a method for automatic evaluation of machine translation. In *in Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, p. 311–318.

SCHWENK H. (2007). Continuous space language models. *Computer Speech and Language*, **21**(3), 492–518.

SCHWENK H. (2012). Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of COLING 2012 : Posters*, p. 1071–1080, Mumbai, India : The COLING 2012 Organizing Committee.

SCHWENK H., R. COSTA-JUSSA M. & R. FONOLLOSA J. A. (2007). Smooth bilingual n-gram translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 430–438, Prague, Czech Republic.

SOCHER R., BAUER J., MANNING C. D. & ANDREW Y. N. (2013). Parsing with compositional vector grammars. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 455–465, Sofia, Bulgaria.

TURIAN J., RATINOV L.-A. & BENGIO Y. (2010). Word representations : A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, p. 384–394, Uppsala, Sweden : Association for Computational Linguistics.

VASWANI A., ZHAO Y., FOSSUM V. & CHIANG D. (2013). Decoding with large-scale neural language models improves translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1387–1392, Seattle, Washington, USA.

WATANABE T., SUZUKI J., TSUKADA H. & ISOZAKI H. (2007). Online large-margin training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* : Citeseer.

YANG N., LIU S., LI M., ZHOU M. & YU N. (2013). Word alignment modeling with context dependent deep neural network. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 166–175, Sofia, Bulgaria.

ZENS R., OCH F. J. & NEY H. (2002). Phrase-based statistical machine translation. In *KI '02 : Proceedings of the 25th Annual German Conference on AI*, p. 18–32, London, UK : Springer-Verlag.