Utilisation de ressources lexicales et terminologiques en traduction neuronale

François Yvon, Sadaf Abdul Rauf

Collection Notes et Documents

UTILISATION DE RESSOURCES LEXICALES ET TERMINOLOGIQUES EN TRADUCTION NEURONALE

François YVON et Sadaf ABDUL-RAUF

JUIN 2020

#2020-01

Utilisation de ressources lexicales et terminologiques en traduction neuronale

François Yvon et Sadaf Abdul-Rauf

Résumé

La traduction automatique (TA) neuronale a conduit à une amélioration perceptible de la qualité de traduction et de l'utilisabilité des textes ainsi produits dans un nombre varié de contextes. Cette technologie repose sur l'exploitation d'algorithmes qui fonctionnent en boite noire, ce qui rend difficile le contrôle fin du processus de traduction. En particulier, alors que la génération antérieure de modèles de traduction (statistique) permettait assez directement d'injecter des ressources dictionnairiques ou terminologiques, l'hybridation de la TA neuronale par des méthodes à base de dictionnaires ou de règles s'avère plus délicate. Ceci est parfois vécu comme une régression, en particulier dans des contextes de traduction assistée par ordinateur (TAO) ou de post-édition (PE), ou encore dans les contextes ou domaines pour lesquels il existe peu de données parallèles.

Dans ce rapport, nous proposons une revue critique des tentatives récentes pour intégrer des lexiques bilingues en TA neuronales, pour constater que la plupart peuvent s'interpréter comme des essais pour adapter au cadre de la TA neuronale des méthodes anciennes. Nous discutons également diverses pistes qui restent à explorer pour rendre cette hybridation de la TA plus prédictible et plus transparente.

Table des matières

1 Contexte

1.1 Introduction rapide à la traduction neuronale

Dans cette section, nous présentons rapidement les principaux concepts de la traduction automatique neuronale (TAN), en nous attardant plus longuement sur ceux dont la compréhension est importante pour décrire avec précision les méthodes d'intégration de ressources linguistiques lexicales. Nous détaillons en particulier le concept d'attention en TAN, ainsi que sur les algorithmes de décodage. Une présentation beaucoup plus complète est donnée dans (Koehn, 2020).

1.1.1 Traduction neuronale : quelques principes généraux

La traduction neuronale se distingue de bien des manières de la génération précédente de systèmes de TA, incarnée par les modèles IBM de Brown *et al.* (1990, 1993), puis par leurs évolutions vers les modèles statistiques à base de segments (Och et Ney, 2002; Koehn *et al.*, 2003; Koehn, 2010), dont l'implantation la plus aboutie est donnée par dans le système Moses (Koehn *et al.*, 2007).

Le principe général de ces architectures reste toutefois le même : celui de produire en langue cible la meilleure traduction $\mathbf{e}$ possible de la phrase source $\mathbf{f}$ en entrée du système, selon une règle de décision probabiliste :

$$\mathbf{e}^* = \underset{\mathbf{e}}{\operatorname{argmax}}\, p_{\boldsymbol{\theta}}(\mathbf{e}|\mathbf{f}), \tag{1}$$

dont les paramètres ($\boldsymbol{\theta}$) sont estimés à partir de corpus parallèles alignés.

L'apprentissage de telles distribution étant irréaliste lorsque l'on utilise des unités discrètes, les modèles de TA statistique reposent sur une approximation prenant la forme [1] suivante :

$$p(\mathbf{e}|\mathbf{f}) = \sum_{\sigma} \prod_{t} p_{\boldsymbol{\theta}}(\mathbf{e}_{\sigma,t}|\mathbf{f}_{\sigma,t}) \approx \max_{\sigma} \prod_{t} p_{\boldsymbol{\theta}}(\mathbf{e}_{\sigma,t}|\mathbf{f}_{\sigma,t})$$

où σ représente toutes les manières de réordonner $\mathbf{f}$ et de le segmenter en segments qui sont synchrones avec ceux de $\mathbf{e}$, et $\mathbf{f}_{\sigma,t}$ (resp. $\mathbf{e}_{\sigma,t}$) représente les segments modélisés qui sont les briques de base du modèle. L'apprentissage vise à estimer les paramètres $\theta_{e,f}$ qui expriment les probabilités de traduction de segments, ainsi qu'un certain nombre de paramètres auxiliaires (pour évaluer la distortion, la probabilité des séquences cibles) qui sont utiles pour résoudre (approximativement) le programme défini par l'équation (1). Un préalable est alors d'aligner mot-à-mot (ou segment-à-segment) les phrases qui constituent le corpus parallèle utilisé pour apprendre ces modèles.

La « révolution » introduite par les méthodes neuronales est essentiellement de rendre tractable la factorisation suivante de la loi conditionnelle :

$$p_{\boldsymbol{\theta}}(\mathbf{e}|\mathbf{f}) = \prod_{t} p_{\boldsymbol{\theta}}(\mathbf{e}_t|\mathbf{e}_{<t}, \mathbf{f}),$$

1. Cette présentation est volontairement idéalisée, on se reportera pour la version détaillée à (Koehn, 2010) ou à (Allauzen et Yvon, 2011) pour une introduction en langue française.

selon laquelle la probabilité de chaque mot est conditionnée par le préfixe courant de la phrase cible ($\mathbf{e}_{<t}$) et l'intégralité de la phrase source ($\mathbf{f}$). La manipulation de telles distributions est rendue possible par la transformation des contextes discrets ($\mathbf{e}_{<t}, \mathbf{f}$) et des mots du vocabulaire $\mathbf{e}_t$ dans des espaces de représentations continus, chaque mot $\mathbf{f}_t$ (et contexte) étant associé à un vecteur numérique en grande dimension noté $E(\mathbf{f}_t)$, qui représente toute l'information utile sur ce mot ou sur ce contexte.

Une telle factorisation suggère que la résolution du programme (1) peut se faire en générant les mots de la gauche vers la droite selon :

$$p_{\boldsymbol{\theta}}(\mathbf{e}_t|\mathbf{f}) = \text{argmax} \prod_t p_{\boldsymbol{\theta}}(\mathbf{e}_t|\mathbf{e}_{<t}, \mathbf{f}). \tag{2}$$

Des éléments complémentaires concernant les algorithmes de décodage dans le cadre de la traduction neuronale sont donnés à la section 1.1.2.

L'estimation de $\boldsymbol{\theta}$ se déroule en maximisant la log-vraisemblance conditionnelle (ou entropie croisée) $\sum_t \log p_\theta(\mathbf{e}_t|\mathbf{e}_{<t}, \mathbf{f})$ accumulée sur un grand ensemble de phrases, donnant lieu à un programme d'optimisation complexe qui est résolu de manière approchée par des algorithmes génériques d'optimisation numérique en grande dimension.

Les principales architectures neuronales se distinguent alors essentiellement selon la manière dont l'encodage du contexte est réalisé : dans les architectures neuronales *récurrentes*, sur lesquelles s'appuient les premiers systèmes de TAN, cet encodage est réalisé par un réseau récurrent (Kalchbrenner et Blunsom, 2013; Cho *et al.*, 2014b), rapidement complété par un *modèle d'attention* (Bahdanau *et al.*, 2014). Les architectures *attentionnelles*, plus récentes, se débarrassent même du composant récurrent (Vaswani *et al.*, 2017a). L'apprentissage d'un tel système devient équivalent à celui d'un classifieur devant produire la classe (le mot) la plus probable étant donné un contexte encodant des informations riches, potentiellement à grande distance.

Les architectures récurrentes Dans les architectures récurrentes initialement proposées par Kalchbrenner et Blunsom (2013); Cho *et al.* (2014b,a), le contexte ($\mathbf{e}_{<t}, \mathbf{f}$) est vu comme une séquence constituée d'une série de mots sources juxtaposée à une séquence de mots cibles. Chaque mot $\mathbf{f}_i$ de la phrase source est associé, au terme d'un encodage plus ou moins complexe (monodirectionnel ou bidirectionnel, à une ou à plusieurs couches), à un vecteur multidimensionnel h_i, le symbole de fin de phrase source étant associé à un état particulier h_I (voir la figure 1).

Le décodeur combine h_I avec le préfixe courant de la phrase cible pour aboutir à une représentation s_j du contexte de prédiction du mot j, qui est alors prédit avec une distribution $p_{\boldsymbol{\theta}}(\mathbf{e}_j|h_I, \mathbf{e}_{j-1})$. Cette approche simpliste, qui résume la phrase source dans un vecteur unique h_I, a été rapidement complétée par Bahdanau *et al.* (2014) dont l'architecture prend en compte un contexte source plus riche noté c_j, qui est recalculé à chaque étape comme une combinaison linéaire convexe de l'ensemble des vecteurs h_i selon :

$$c_j = \sum_i \alpha_{ji} h_i, \text{ avec } \alpha_j = \text{softmax}(e_j), e_{ji} = (V_a^T \tanh(W_{as} s_{j-1} + W_{ah} h_i))$$

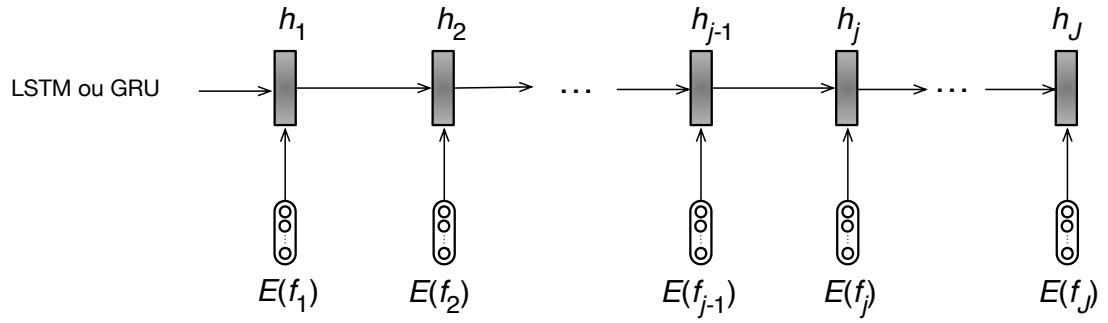

FIGURE 1 – Encodage de la phrase source

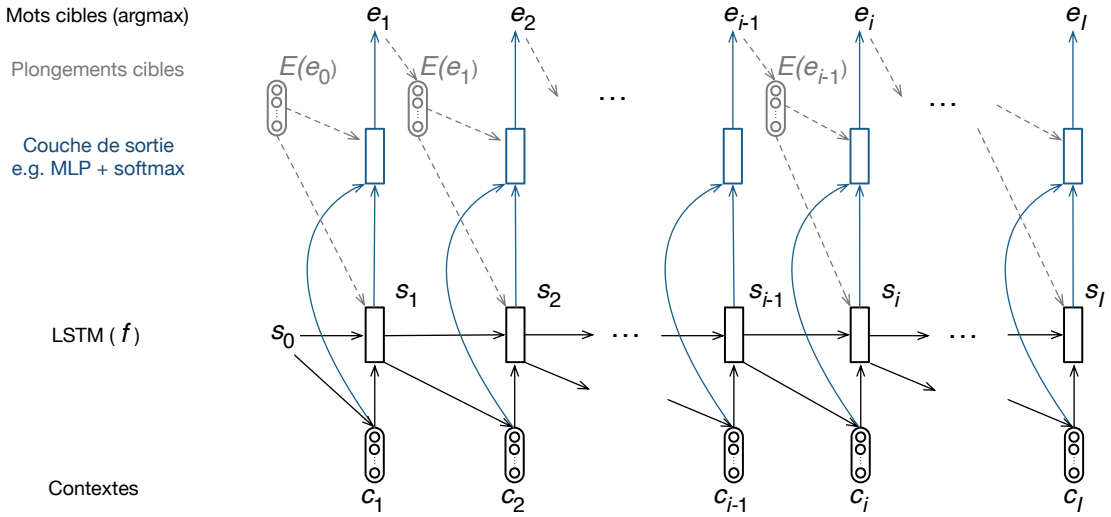

FIGURE 2 – Décodage de la phrase source, avec attention

Les coefficients α_{ij} mesurent une affinité normalisée entre le contexte cible courant résumé dans s_{j-1} et chacun des mots sources $\mathbf{f}_i$ sur la base de sa représentation h_i. L'ensemble des vecteurs $\{\boldsymbol{\alpha}_j, j = 1 \dots J\}$ forme une matrice stochastique qui quantifie, pour chaque pas de temps de la traduction, l'importance locale des mots sources dans la décision. Le décodage exploite alors $p_{\boldsymbol{\theta}}(\mathbf{e}_t|\mathbf{e}_{<t}, \mathbf{f}) = p_{\boldsymbol{\theta}}(\mathbf{e}_t|s_t, c_t)$ (voir la figure 2).

Si la matrice d'attention peut être assimilée en première approche à une matrice d'alignements (probabilistes) entre les mots cibles et les mots sources, de nombreux travaux ultérieurs (par exemple (Cohn *et al.*, 2016; Koehn et Knowles, 2017; Ghader et Monz, 2017)) ont montré qu'en l'absence de contraintes supplémentaires sur la structure de cette matrice, les valeurs qu'elle contient diffèrent assez fortement des valeurs produites par les aligneurs probabilistes classiques tels que les modèles IBM (Brown *et al.*, 1993; Och et Ney, 2003). De cette observation ont découlé de nombreuses tentatives pour soit superviser le calcul de l'attention par des alignements de référence, soit contraindre la

matrice d'attention à ressembler d'avantage à une matrice d'alignement (en limitant la « fertilité », en imposant des contraintes de distortion, etc).

Les architectures purement « attentionnelles » Les architectures récurrentes posent un problème majeur à l'apprentissage : le calcul de la fonction de perte qui sert de fondement à l'estimation des paramètres doit être effectue de manière séquentielle, puisque sa valeur à l'instant t dépend récursivement des représentations calculées aux instants précédents : s_t dépend de s_{t-1}, qui dépend de de s_{t-2} ainsi que de tous les choix de traduction déjà effectués. L'architecture Transformer de Vaswani *et al.* (2017b) permet de pallier ce problème, le préfixe $\mathbf{e}_{<t}$ étant traité dans la décision de la même manière que la source, en calculant une *auto-attention* qui rend toutes les positions précédentes $t-1, t-2..., 1$ également importantes dans la sélection du mot courant. On peut ainsi voir le modèle Transformer comme combinant simplement un encodeur attentionnel et un décodeur attentionnel.

Les détails de l'architecture ne sont pas essentiels ici, si ce n'est que (a) elle permet de calculer simultanément la contribution à la fonction objectif de tous les mots cibles ($\{-\log p_{\boldsymbol{\theta}}(e_t|\mathbf{e}_{<t}, \mathbf{f}), t = 1 \dots I\}$), qui permet des implantations efficaces sur GPU. Elle donne lieu également au calcul d'une matrice d'attention [2] source-cible qui peut elle aussi être interprétée, manipulée ou entraînée comme une matrice d'alignement. Comme pour les modèles récurrent, l'équivalence entre attention et alignement n'est pas directe (Li *et al.*, 2019b; Ding *et al.*, 2019). Dans cette architecture, le décodage doit se dérouler de gauche à droite, puisque chaque préfixe déjà construit conditionne la génération des mots futurs.

1.1.2 Principes du décodage neuronal

Recherche gloutonne Le processus de génération de la phrase cible ou *décodage* se déroule en général de la gauche vers la droite reproduisant l'ordre naturel de l'écriture (dans les langues indo-européennes). L'approche *gloutonne* (*greedy*) produit à chaque pas de temps t le mot $\mathbf{e}_t$ (ou le token [3]) le plus probable parmi tous les mots du vocabulaire cible V_e conditionnellement au préfixe courant (en langue cible, noté $\mathbf{e}_{<t}$) et à la phrase source, selon :

$$\mathbf{e}_t^* = \underset{\mathbf{e}_t \in V_e}{\operatorname{argmax}}\, p_{\boldsymbol{\theta}}(\mathbf{e}_t|\mathbf{e}_{<t}, \mathbf{f}).$$

Dans la version gloutonne, le décodage s'arrête dès que le système émet le symbole de fin de phrase </s>. Cette méthode se base sur des décisions prématurées et peut conduire

2. Plus précisément, d'un ensemble de matrices d'attentions, puisqu'il y en a une par couche de l'encodeur.

3. Pour pouvoir traiter de vocabulaires ouverts, les systèmes de TA s'appuient sur un découpage des mots en unités sous-lexicales calculées sur des bases purement statistiques (Sennrich *et al.*, 2016b). Ces unités ont l'immense avantage de permettre de traiter de manière homogène des langues variées, mais conduisent à faire disparaitre la notion de mots qui est pourtant centrale pour accéder aux ressources et modèles linguistiques traditionnels.

à des erreurs de recherche, qui correspondent aux situations où le décodeur échoue à trouver $\arg\max_{\mathbf{e}\in V_e^\star} p_\theta(\mathbf{e}|\mathbf{f})$.

Contraintes de couverture L'algorithme de décodage glouton esquissé ci-dessus diffère des algorithmes de décodage utilisés en traduction statistique en particulier par leur apparente absence de synchronisation entre les mots cibles et leurs équivalents sources. En résultent un certain nombre de problèmes identifiés très tôt, par exemple la propension des décodeurs neuronaux à omettre la traduction de mots sources, ou au contraire, à en traduire certains plusieurs fois. La solution de Tu *et al.* (2016) consiste à réintroduire des *contraintes de couverture*, visant à maintenir pendant le décodage la connaissance des mots traduits ou restant à traduire. Divers modèles sont comparés dans cette étude, qui s'appuient sur des modifications du mécanisme d'attention : il s'agit essentiellement d'injecter une mémoire des vecteurs d'attention passés dans le calcul de l'attention courante, en faisant en sorte que le calcul du vecteur d'attention c_t dépende des valeurs passées $c_{<t}$ (normalisées ou non), ainsi que de la fertilité de chaque mot-source (certains pouvant demander plus d'attention que d'autres) (Cohn *et al.*, 2016). Les auteurs concluent que cette méthode permet d'améliorer la traduction en particulier des phrases longues, pour lesquelles ces questions de sous/sur traduction sont le plus patentes. Une méthode alternative, qui réexprime ces mêmes contraintes sous la forme de plongements numériques, est présentée par Mi *et al.* (2016a).

Les expériences décrites par Wu *et al.* (2016) explorent une autre manière de contrôler la couverture et la longueur pendant le décodage, en ordonnant les candidats-traductions selon un score qui vise à (a) normaliser les hypothèses par longueur et (b) s'assurer que chaque mot source a été sélectionné par le module d'attention. Ils en donnent les expressions suivantes :

$$
\begin{aligned}
s(\mathbf{f}, \mathbf{e}_{1:t}) &= \frac{\log p_{\boldsymbol{\theta}}(\mathbf{e}_t|\mathbf{e}_{<t}, \mathbf{f})}{\mathrm{lp}(\mathbf{e}_{1:t})} + \mathrm{cp}(\mathbf{e}_{1:t}) \\
\mathrm{lp}(\mathbf{e}_{1:t}) &= \frac{(5+t)^\gamma}{(5+1)^\gamma} \\
\mathrm{cp}(\mathbf{e}_{1:t}) &= \delta \sum_{i=1}^{I} \log(\min(\sum_{j=1}^{t} \alpha_{ji}, 1),
\end{aligned}
$$

avec α les poids d'attention et γ et δ des paramètres fixés sur un corpus de développement.

Recherche en faisceau La recherche en faisceau (*beam search*) est une méthode de recherche heuristique qui étend la méthode gloutonne en conservant à chaque instant un ensemble de préfixes actifs $B_t = \{\mathbf{e}_{<t,k}, k = 1 \dots B\}$. À chaque pas de temps, les successeurs possibles de ces B préfixes sont évalués et un nouvel ensemble B_{t+1} en est déduit. Il existe deux grandes familles d'approches pour développer B_{t+1} :

— soit conserver tout préfixe dont la probabilité cumulée n'est pas « trop éloignée » du meilleur préfixe $\mathbf{e}_{<t,1}$, en conservant toute hypothèse $\mathbf{e}_{<t,1}$ telle que $p_\theta(\mathbf{e}_{<t,k}|\mathbf{f}) >$

$(1-\alpha)p_\theta(\mathbf{e}_{<t,1}|\mathbf{f})$, avec $\alpha \in [0,1]$ un paramètre qui contrôle la largeur du faisceau. Cette manière de procéder conduit à maintenir un nombre variable de préfixes actifs ;

— soit conserver les B meilleurs préfixes, ce qui présente le mérite de maintenir un nombre de préfixes actifs constant. Cette variante est également connue sous le nom d'élagage par histogramme (*histogram pruning*).

Dans la recherche en faisceau, le décodeur s'arrête dès que :

— un préfixe actif correspond à une phrase complète (finissant par </s>) qui ne pourra plus être développée ;

— tous les autres préfixes actifs ont un score inférieur à la phrase complète et ne pourront donc le surpasser dans le futur.

La complexité de cet algorithme est $O(JBV_e)$, puisqu'à chaque pas de temps on calcule les V_e continuations possibles des B préfixes.

Erreurs de recherche La mise en œuvre de la recherche en faisceau s'est longtemps heurtée à un paradoxe apparent, selon lequel augmenter la taille de B, donc l'espace de recherche, conduisait à résultats dégradés. Diverses explications ont été mises en avant (Murray et Chiang, 2018; Stahlberg et Byrne, 2019), la plus convaincante se fondant sur les constatations suivantes : (a) faute d'intégrer des informations sur la longueur de la source, le critère d'arrêt du décodage est l'émission par le décodeur d'un symbole de fin de phrase (</s>) ; (b) la probabilité des séquence est un produit, et celle des séquences courtes est en générale meilleure que celles des séquences longues. Augmenter l'espace de recherche conduit à inclure dans le faisceau des séquences trop courtes, qui vont pourtant s'avérer les plus probables pour le décodeur.

Pour l'illustrer par un cas extrême, soit k le rang de l'hypothèse qui génère </s> au premier pas de temps et conduit donc à une traduction vide ; si $B \geq k$, alors cette hypothèse entre dans le faisceau à l'instant $t = 1$ et son score ne changera plus, alors que toutes les autres hypothèses concurrentes verront leur score décroître au cours de leur développement, conduisant souvent cette hypothèse à être finalement la préférée. Normaliser le score des hypothèses par leur longueur, comme proposé par Wu *et al.* (2016) (voir supra) permet de rendre les hypothèses au sein du faisceau plus comparables entre elles.

1.1.3 La question de l'adaptation au domaine

La traduction automatique, comme tout système fondé sur l'apprentissage automatique, repose sur l'hypothèse que les données disponibles pour l'apprentissage sont similaires à celles qui seront l'objet des traductions futures, soit, en termes statistiques, qu'elles sont tirées sous la même distribution $\mathcal{D}$. Lorsque la distribution des données d'apprentissage $\mathcal{D}_a$ diffère de la distribution des données de test $\mathcal{D}_t$, les paramètres estimés $\boldsymbol{\theta}$ ne sont plus nécessairement optimaux pour traduire de nouvelles données et se pose la question de *l'adaptation au domaine*.

Ce terme général recouvre une gamme importante de situations correspondant à des différences de genre, de registre, de domaine ou de style qui se manifestent par des va-

riations statistiques dans les usages des termes ou dans la distribution des constructions syntaxiques dans les textes sources, pouvant donner lieu à des variations dans les traductions associées. Pour prendre un exemple caractéristique : la traduction de l'anglais *chair* lorsqu'on l'apprend sur des documents issus du corpus Europarl est *président*, qui n'est pas la meilleure traduction lorsque l'on s'intéresse à la traduction d'autres types de textes. Un cas extrême, important en pratique, est celui où un vocable (ou un terme, ou un sens), n'est jamais observé dans le corpus d'apprentissage et doit malgré tout être traduit. Il est donc admis que changer de domaine conduit à des erreurs dans la traduction des termes ou des vocables spécialisés (Irvine *et al.*, 2013) et que l'adaptation au domaine peut, dans une certaine mesure, aider à pallier ces défaillances.

Il existe une littérature foisonnante sur ces sujets, qui sont bien documentés en traduction statistique comme en traduction neuronale (Chu *et al.*, 2017; Chu et Wang, 2018). Plusieurs stratégies sont typiquement considérées selon le contexte dans lequel l'apprentissage se déroule. Lorsque des données parallèles du domaine source sont disponibles, même en faible quantité, la démarche commune consiste à spécialiser un modèle générique apprise sur des données hors-domaine. On parle dans ces situations *d'adaptation supervisée* ou de *fine-tuning* (Luong et Manning, 2015; Freitag et Al-Onaizan, 2016). Plutôt que mélanger les données, une alternative est d'opérer une combinaison entre domaines au niveau des règles de décision (en utilisant un mélange probabiliste de plusieurs modèles (Foster et Kuhn, 2007)), au niveau des paramètres, des états internes ou encore au niveau des sorties discrètes.

Une situation moins favorable est celle où l'on ne dispose que d'exemples non traduits (en langue source ou en langue cible) de textes du domaine d'intérêt. Dans le premier cas, on pourra par exemple essayer de pondérer les données hors-domaine en fonction de leur pertinence pour le domaine cible (Axelrod *et al.*, 2011; Duh *et al.*, 2013), en filtrant celles qui en sont le plus éloignées, ou en surpondérant celles qui en sont le plus proches. Dans le second cas, deux stratégies sont également envisageables. La première consiste à estimer un modèle de langue statistique (ou neuronal) de la langue cible pour le domaine d'intérêt, qui est combiné au contexte courant dans la règle de décision (2). Cette stratégie est explorée, par exemple, dans (Gulcehre *et al.*, 2017; Stahlberg *et al.*, 2018a) : deux méthodes de fusion sont considérées : l'une qui utilise la probabilité du modèle de langue pour réévaluer les hypothèses proposées par le décodeur neuronal, avec une combinaison simple linéaire entre les deux termes (Gulcehre *et al.* (2017) parle de *fusion superficielle* (*shallow fusion*)), l'autre qui recombine les états internes du décodeur et du modèle de langue avant la génération des hypothèses lexicales : on parle alors de *fusion profonde* (*deep fusion*). Ces deux approches sont illustrées sur la figure 3.

La seconde consiste à rétrotraduire automatiquement, pour autant qu'on dispose des outils de traduction idoines, ces textes cibles en langue source, puis à utiliser ces données artificielles comme des données d'adaptation supervisée (Sennrich *et al.*, 2016a). Dans les deux cas l'intention est la même : corriger les distributions lexicales et syntaxiques des textes cibles en prenant en intégrant dans le calcul de la fonction objectif des exemples du domaine d'intérêt. Cette méthode est étudiée en détail dans (Poncelas *et al.*, 2018; Burlot et Yvon, 2018).

L'adaptation au domaine est une étape cruciale pour tirer le meilleur parti des res-

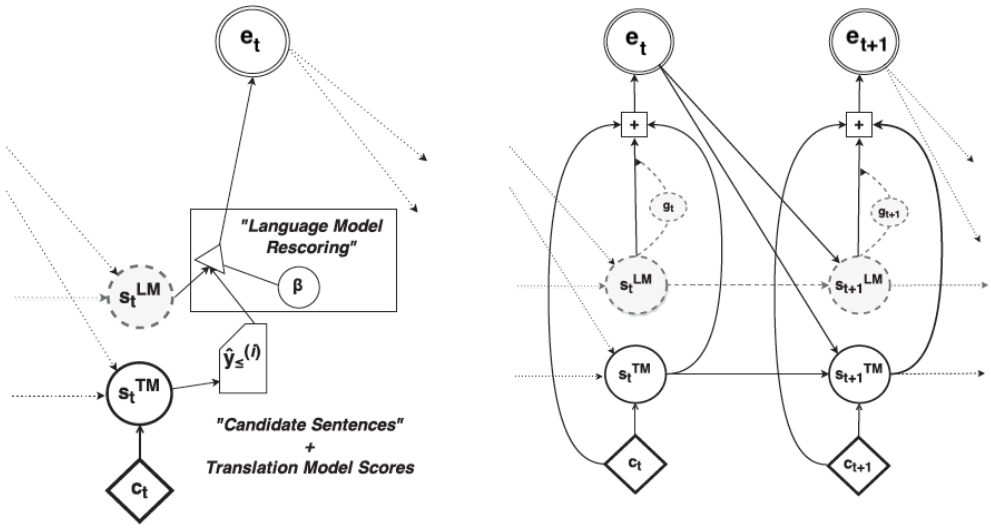

FIGURE 3 – Deux méthodes pour fusionner modèle de langue et décodage neuronal : fusion superficielle (post-génération) à gauche ; fusion profonde (pré-génération) à droite. Figure extraite de Gulcehre *et al.* (2017).

sources disponibles et constitue donc une technique à utiliser impérativement pour traduire dans les domaines de spécialité. Pour autant, elle ne constitue pas une réponse complètement satisfaisante aux questions abordées dans ce rapport, puisqu'elle n'apporte aucune garantie concernant la traduction correcte des termes du domaine. Tout au plus peut on en espérer une amélioration des représentations lexicales et une meilleure prédiction des séquences typiques de la langue cible (Scansani *et al.*, 2019), pour autant qu'elles ne soient pas concurrencées par des traductions alternatives surreprésentées à l'apprentissage. Cette littérature ne sera donc pas présentée davantage dans ce rapport et le lecteur intéressé de ces questions pourra se reporter aux références données ci-dessus ou à la revue de la littérature présentée dans (Chu *et al.*, 2017).

1.2 Ressources lexicales en traduction automatique

Cette section explicite les contextes dans lesquels il peut être souhaitable de contrôler les choix lexicaux d'un système de traduction automatique neuronale, les types d'unités qui sont contrôlées et les problèmes que posent ce contrôle. On notera que d'autres types de contrôle de la TA pourraient être ou ont été explicitement considérés dans la littérature, par exemple pour imposer un niveau de formalité (Niu *et al.*, 2017), un niveau de complexité (Agrawal et Carpuat, 2019; Marchisio *et al.*, 2019), voire un contrôle plus explicite de la structure syntaxique des phrases générées. Ces questions ne sont pas détaillées plus avant ici.

1.2.1 Contrôler les traductions neuronales

Le besoin d'opérer un contrôle direct des sorties d'un système de traduction statistique peut apparaitre dans plusieurs contextes :

1. pour améliorer la traduction des mots rares ou hors-vocabulaire, c'est-à-dire n'apparaissant pas dans le corpus parallèle (Arthur *et al.*, 2016). Si l'on peut penser que pour les langues bien dotées cette question est résolue par l'utilisation d'unités sous-lexicales (Sennrich *et al.*, 2016b; Kudo et Richardson, 2018), voire de modèles de caractères (Costa-jussà et Fonollosa, 2016; Luong et Manning, 2016), elle reste importante pour les paires de langues moins bien dotées en ressources parallèles ;
2. pour traduire correctement les termes, idiomes et expressions polylexicales dont les équivalents en cible ne peuvent être construits compositionnellement (Hasler *et al.*, 2018) ;
3. pour garantir qu'un certain nombre de vocables essentiels à la correction (*adequacy*) de la traduction, mais qui par essence ou par manque de données spécialisées ne seront jamais tous observés (ou suffisamment observés) dans les corpus d'apprentissage, sont correctement traités. Il s'agit en particulier des extra-lexicaux : nombres, montants, dates, noms propres (de lieux, de personnes, d'organisations, de marques), adresses, URL, ou encore *hashtags*, avec des spécificités selon les domaines. Ainsi dans le domaine de la finance, des noms propres de titres ou de fonds, ou dans le domaine médical, des noms d'entités chimiques ou biologiques, dans le domaine du droit, des entités juridiques (lois, décrets, règlements) ou des acteurs particuliers du système juridique. La préoccupation peut ici se limiter au bon traitement des phrases isolées.
4. pour s'assurer que les choix de traduction restent cohérents tout au long du document (Carpuat et Simard, 2012) et qui demandent d'excercer un contrôle entre les différentes phrases d'un même texte (Meng *et al.*, 2014). Cette démarche s'inscrit dans une démarche plus générale de transparence et d'explicabilité du système de traduction, dont on peut attendre (a) qu'il respecte les choix de traductions imposés par un client ou ceux que préférera un réviseur ou un post-éditeur ; (b) qu'il explicite, par exemple par le truchement d'un alignement, les raisons pour lesquelles tel ou tel terme se retrouve dans la sortie (Stahlberg *et al.*, 2018b).

Pour les situations [1] et [2] supra, les ressources privilégiées correspondent à des listes finies de correspondances entre unités source et cibles et posent principalement les problèmes suivants :

— quelles informations contiennent ces listes ? Lemmes ou formes fléchies, contexte d'occurence, probabilités de traduction, etc.
— comment les recueillir ? À partir de l'exploitation de corpus bilingues, parallèles ou comparables, ou par exploitation de ressources construites manuellement (dictionnaires, terminologies, etc) ;
— comment détecter les contextes sources dans lesquels la traduction doit être imposée ? Le problème se pose un peu différemment pour les dictionnaires « généraux »

contenant des mots simples, pour lesquels la question de la désambiguisation sémantique est essentielle, et pour les terminologies qui comprennent de nombreuses unités polylexicales qui sont moins ambigües, mais qui peuvent correspondre à des segments discontinus. Une seconde question concerne la variation de forme en langue source, qui peut être due soit aux processus morphologiques actifs en source, soit au caractère bruité des énoncés à traduire (non respect de la typographie, typos et fautes d'orthographes, etc). Notons enfin que pour ce qui concerne les unités polylexicales, d'autres formes de variation sont possibles : insertion de mots rendant le terme discontinus, inversions, transformations morphologiques etc ;

— comment insérer le segment cible dans l'hypothèse de traduction tout en préservant sa correction ? Cette question se décompose en deux problèmes un peu différents : celui de la position du segment inséré, et celui des marques morphologiques qu'il devra porter. L'intensité de ces deux problèmes varie selon les langues en fonction de la complexité des processus de morphologiques impliqués.

La situation [3] est un peu différente, car le nombre des unités à contrôler n'est pas fini et ne se prête pas à une approche par énumération : qu'on pense par exemple aux noms propres ou encore aux adresses mail ou aux URL. Il en va naturellement de même pour les traductions cibles, qui devront être produites automatiquement, prenant souvent par exemple la forme d'une recopie du segment source. Les ressources privilégiées correspondent alors à des expressions rationnelles ou *regexp* visant à identifier ces unités particulières côté source. Les questions qui se posent alors sont :

— comment repérer avec les entités sources à contrôler, en particulier en présence de variabilité (morphologique, typographique), ou de motifs se superposant (par exemple un nom propre enchâssé dans un autre nom propre) ?

— quelles traductions leur associer au delà de la recopie directe ? Se posent dans ce contexte les problèmes classiques de la *localisation* : transformation des unités de mesure, des expressions numériques et temporelles, translittération des noms propres lorsque les systèmes d'écriture source et cible sont différents (Grundkiewicz et Heafield, 2018), etc.

— comment les insérer dans le contexte cible avec, comme précédemment, la possibilité que ces unités puissent porter des marques morphologiques en fonction de leur contexte. C'est par exemple le cas des nombres ou des noms propres en russe qui portent des marques flexionnelles.

La situation [4] renvoie à des besoins un peu différents, pour lesquels les unités à contrôler ne sont pas définies à priori mais varient (d'un client à l'autre, d'un traducteur à l'autre, voire d'un document à l'autre). Les questions à résoudre demeurent toutefois globalement les mêmes : comment acquérir ces contraintes ? comment repérer les unités source ? comment sélectionner leur traduction ? comment les fléchir et les insérer au mieux dans leur contexte source ?

Du point de vue des ressources, deux situations principales émergent donc : celles où le fragment source est décrit *en intension* (par une expression régulière ou un programme) et où le fragment cible en sera la recopie - par exemple dans le cas des expressions numériques ; celles où le fragment source est décrit en extension, sous la forme d'un dictionnaire bilingue appariant (au niveau des lemmes, au moins) les vocables sources et

Unités	Disponibilité	Repérage Auto ?	Contexte Requis ?	Traduction Variable ?
mots simples	dictionnaires, corpus	matching + variantes formelles	oui	oui
mots composés, segments	dictionnaires, corpus	matching + variantes formelles	±	oui
termes	dictionnaires, corpus	matching + variantes formelles	non	±
idiomes, expressions figées	dictionnaires, corpus	matching + variantes formelles et syntaxiques	±	oui
noms propres	listes + regexp	matching	non	non (copie)
extra-lexicaux	regexp	matching	non	non (copie)

TABLE 1 – Contraintes lexicales en traduction automatique

cibles : dans cette situation, la recopie n'est plus de mise, et il faut substituer en langue cible le segment identifié par sa traduction de référence, le cas échéant en effectuant une adaptation morphologique idoine.

De manière abstraite, on dénotera sous le nom général de *contraintes* les unités sources dont on souhaite imposer une traduction particulière dans la cible et on supposera qu'elles prennent la forme (idéalisée) d'une réécriture inconditionnelle : (segment source → segment cible). Certaines méthodes sont de surcroit capables de prendre en compte la vraisemblance de la réécriture (hors contexte ou en contexte) ; on peut penser que d'autres informations auxiliaires (le contexte de validité de la traduction, les variations morphologiques autorisées) pourraient également s'avérer très précieuses dans ce contexte, même si elle sont plus difficiles à recueillir.

1.2.2 Intégration de ressources lexicales en traduction statistique

Le besoin de mieux contrôler le comportement d'un système neuronal, en particulier du point de vue des traductions lexicales, n'est pas propre à la traduction neuronale et les mêmes questions se sont posées, peu ou prou, pour les systèmes statistiques. Pour ce qui concerne le repérage de la source, mentionnons par exemple (Carpuat et Diab, 2010). Pour ce qui concerne l'insertion de la traduction, la réponse la plus élaborée en la matière consiste à intervenir dans le décodeur pour introduire dans la phrase cible les traductions désirées et est implantée dans le système Moses (Koehn *et al.*, 2007). Ce fonctionnement ayant motivé ou inspiré de nombreux travaux ultérieurs qui visent essentiellement à le mettre en œuvre dans un cadre neuronal, nous le décrivons sommairement ci-dessous.

Le mécanisme de pré-traduction du système Moses insère dans la source une pré-traduction de segments dont on souhaite contrôler explicitement la traduction, en les encapsulant dans des balises comme dans l'exemple suivant :

das ist ein kleines <n translation="dwelling||house" prob="0.8||0.2">haus</n>',

qui contraint la traduction de l'allemand « haus » à utiliser soit la traduction « dweling » (avec probabilité 0.8), soit la traduction « house » (avec probabilité 0.2). Plusieurs modes de fonctionnement sont possibles : soit forcer la traduction de manière déterministe, soit laisser le décodeur choisir entre la prétraduction et les hypothèses qui dérivent de la table de traduction, soit encore limiter les choix du décodeurs aux hypothèses compatibles avec les contraintes [4].

Pour être complets, mentionnons que plusieurs autres stratégies pour exploiter de telles ressources ont été proposés en TA statistique [5] :

— la prise en compte de dictionnaires comme un corpus parallèle supplémentaire. Notons que pour l'apprentissage le même résultat serait obtenu (aux statistiques près) en introduisant directement l'entrée bilingue dans la table des segments ;
— l'utilisation de traits (features) dans la table des segments pour identifier les traductions non compositionnelles (Carpuat et Diab, 2010) ;
— l'exploitation du contexte large au niveau document (par exemple via des modèles thématiques) pour influencer la traduction de termes (par exemple (Meng *et al.*, 2014)) ;
— l'introduction de termes ou de dictionnaires comme un modèle de langue supplémentaire pour fournir une mémoire à court terme (à la manière des modèles de cache de (Kuhn et DeMori, 1990)), voir par exemple (Bertoldi *et al.*, 2013) dans un contexte de traduction assistée par ordinateur.

Plusieurs de ces approches sont comparées dans Bouamor *et al.* (2012), en utilisant des expressions poly-lexicales extraites du corpus Europarl pour la paire français-anglais. Comme on le verra ci-dessous, ces méthodes ont également servi d'inspiration pour injecter des connaissances dans un contexte de TA neuronale.

1.3 Vers un meilleur contrôle des traductions neuronales : une vue d'ensemble

Intégrer des lexiques de traduction est donc relativement aisé en traduction statistique, à cause des contraintes de couverture, qui impliquent qu'à chaque pas de temps, le décodeur est informé du segment source en cours de traduction. Imposer une traduction particulière pour les segments d'intérêt peut alors être réalisé avec 100% de certitude que la contrainte sera appliquée.

Le décodage opéré par les systèmes neuronaux n'intègre en général pas de contrainte de couverture [6], ce qui fait que l'information concernant le segment en cours de traduction n'est pas explicitement disponible. Plusieurs manières de procéder sont alors envisageables, qui réalisent des compromis un peu différents entre d'une part les garanties obtenues et le surcroit de complexité du décodage induit par l'introduction de contraintes. Nous les présentons dans les sections suivantes en les ordonnant de la moins à la plus contrainte :

4. Voir http://www.statmt.org/moses/?n=Advanced.Hybrid#ntoc1
5. Nous ne visons pas ici à l'exhaustivité.
6. Ce qui est parfois un problème, voir (Tu *et al.*, 2016).

(a) marquer explicitement la présence dans la source d'unités d'intérêts par des tokens spéciaux, en espérant qu'ils seront automatiquement réinsérés dans la phrase cible, ce qui permettra leur réécriture par post-traitement. Cette méthode est simple et n'altère pas le système de traduction, mais les garanties sont faibles, et les risques d'échec importants. Ils sont discutés à la section 2.

(b) prétraduire les segments d'intérêts en modifiant la phrase source fournie au système de manière à ce qu'elle incorpore déjà la traduction désirée. Pour cette approche, le système doit savoir traiter d'énoncés composites (voir la section 2.4).

(c) introduire les contraintes comme des indices contextuels complémentaires, à la manière dont on peut utiliser un modèle de la langue cible ; c'est par exemple l'approche de Feng *et al.* (2017), qui est discutée à la section 3.3 ;

(d) imposer que pour chaque segment d'intérêt détecté dans la phase source, la traduction de référence que l'on souhaite imposer se trouve dans la cible. Cette stratégie altère le fonctionnement du décodeur et peut le ralentir sensiblement. En revanche, elle apporte des garanties fortes sur la présence du segment cible, sans contrôler toutefois le contexte dans lequel il est introduit. Cette approche est en particulier retenue par Hokamp et Liu (2017); Post et Vilar (2018), elle est présentée en détail dans la section 3 ;

(e) simuler un décodage à la façon d'un système statistique, en recalculant les informations de couverture à partir des informations fournies par le module d'attention : l'information concernant le segment source en cours de traduction redevient disponible et on peut en contraindre le résultat aux positions choisies. C'est, par exemple, la démarche suivie par Chatterjee *et al.* (2017); Alkhouli *et al.* (2018); Song *et al.* (2020), qui sera discutée à la section 4.

2 Contrôle de la boite noire : pré-et post-traitements

Plusieurs méthodes visant à contrôler la traduction de formes spécifiques dans la sortie passent par un pré-traitement de l'entrée, qui est ensuite traitée de manière conventionnelle. Nous les recensons ci-dessous en indiquant les limites des solutions proposées.

2.1 Utilisation de corpus artificiels

2.1.1 Rétro-traduction et recopie

La recopie *verbatim* de mots sources inconnus est souvent la meilleure stratégie pour traiter en particulier les extra-lexicaux. L'étude de Currey *et al.* (2017) propose une méthode très simple pour renforcer la propension du décodeur à opérer des recopies. Elle consiste à augmenter le corpus d'apprentissage avec des pseudo phrases-parallèles composées de deux phrases cibles (identiques). Leur analyse, qui porte sur les paires de langue anglais-turc et anglais-roumain, montrent un effet sur la traduction des entités nommées. Nonobstant la question (difficile) du bon équilibrage des phrases vraiment parallèles avec les phrases artificielles, cette manière de procéder ne fournit que très peu de contrôle sur les conditions dans lesquelles la recopie sera ou non effectuée. Burlot

(2019) étudie également cette méthode en comparant plusieurs méthodes pour contrôler la présence des termes d'intérêt dans la rétro-traduction pour une tâche de traduction français-allemand. La création de corpus artificiels mélangeant (côté source) des données multilingues est également envisagée à la section 2.4.

2.1.2 Dictionnaires et corpus

Une approche la plus simple pour injecter des dictionnaires bilingues en traduction automatique à base de corpus consiste simplement à considérer les entrées dictionnairiques comme des phrases parallèles, et est documentée par exemple dans (Tan *et al.*, 2015) (et dans les références citées dans ce travail) sous le terme d'utilisation *passive* d'un dictionnaire.

Les travaux de Zhang et Zong (2016) en proposent une version un peu modifiée pour la traduction neuronale, qui consiste à insérer les entrées dictionnairiques dans des phrases parallèles artificielles engendrées à l'aide d'un système statistique – une méthode qui se rapproche d'une forme de *rétro-traduction* (cf. la section 1.1.3) contrôlée. Cette approche a pour elle sa simplicité, puisqu'elle ne demande aucune adaptation du système neuronal, mais pêche par le manque de garanties qu'elle donne, puisque rien n'assure que le système utilisera cette information supplémentaire. Par ailleurs, le choix des phrases artificielles et leur nombre reste une question largement empirique. Ces deux approches sont comparées dans (Rikters et Bojar, 2017).

2.2 *Placeholders* et masques

Principes Crego *et al.* proposent de repérer dans la phrase source les mots dont on souhaite contraindre la traduction. Un prétraitement remplace alors ces formes par un token générique (un *placeholder* ou *masque*, dans les termes de Post *et al.* (2019)) : *j'ai 10 ans* → *j'ai [nombre] ans*. En faisant l'hypothèse que ces tokens génériques se retrouvent côté cible, il s'agira ensuite de les remplacer par la traduction souhaitée dans une étape de post-traitement : *I am [nombre] years old* → *I am 10 years old*. Cette approche implique donc trois étapes : masquage - traduction - démasquage, et peut être retracée, via (Luong *et al.*, 2015), à des travaux anciens en modélisation des langues utilisant des étiquettes de classe pour remplacer les mots rares.

Sont remplacées des expressions qui sont typiquement non traduites, ou bien dont la traduction peut s'opérer par règles, telles que les nombres, les noms de personnes, les lieux, les URL ou encore les expressions temporelles (date, heure, etc). Pour prendre en compte la possibilité qu'une phrase requière simultanément plusieurs masques, ces masques sont de surcroit nantis d'un indice, alors que Luong *et al.* (2015) utilise des informations d'alignement ; Post *et al.* (2019) montre que l'appariemment entre masque source et masque cible peut également se faire automatiquement pendant le post-traitement.

Une telle méthode garantit que la traduction souhaitée apparaîtra dans la sortie, pourvu que le token générique ait bien été transmis vers la cible. Elle permet également de gérer aussi bien les copies forcées (*10* se traduit en anglais *10*) que les traductions contraintes (*Londres* se traduit *London*) au moyen d'un lexique bilingue. Des modules de

translittération peuvent également être employés lorsque les paires de langues concernées le demandent (*Пушкин* se transcrit *Pouchkine* en français). C'est l'approche poursuivie par Li *et al.* (2018), qui construit un dictionnaire bilingue de noms propres chinois-anglais en utilisant un système neuronal à base de caractères.

Limitations du masquage L'inconvénient majeur de cette approche réside dans l'utilisation de deux façons de traduire qui sont entièrement indépendantes, l'une fondée sur des connaissances à priori définies hors-contexte, l'autre qui s'appuie sur le décodage neuronal, et sont recombinées aveuglément dans la sortie. Le système de traduction neuronal ne dispose d'aucune information concernant la traduction contrôlée, puisqu'il n'a accès qu'à un token générique qui ne contient, par définition, qu'une quantité minimale d'information sur ce qui se trouvait dans la source. Par exemple, l'introduction d'un token générique *[nombre]* en russe empêche le système d'engendrer le cas correct, puisque les nombres peuvent porter des marques casuelles qui varient en fonction de la position syntaxique du nombre et de sa valeur. Ignorer le nombre qui était présent dans la source interdira de pouvoir prédire le bon cas.

De même, sauf à pouvoir engendrer statiquement l'ensemble des variantes morphologiques des traductions des entités à contrôler, incluant par exemple tous les cas, nombres, ou marques de déterminations possibles, il n'est pas possible de s'assurer que l'entité copiée en cible sera correctement fléchie en fonction de son contexte d'insertion. Au-delà des stratégies de masquage, ce problème se pose à toutes les approches qui manipulent des lexiques de lemmes.

Un autre inconvénient est lié à la façon dont ces tokens génériques sont repérés dans une phrase source. Par exemple le repérage d'entités nommées suppose la mise en place d'un étiqueteur indépendant du système de traduction, ce qui peut affecter négativement le temps de décodage.

L'analyse de Post *et al.* (2019) souligne que la mise en œuvre de cette approche repose sur deux hypothèses : d'une part, que le décodeur produira exactement le bon nombre (avec un indice correct) de masques dans la cible qu'il en aura rencontrés dans la source, ou que l'appariemment pourra être réalisé en post-traitement ; d'autre part qu'il ne possédait pas déjà la connaissance nécessaire à réaliser correctement ces traductions. Une analyse détaillée du fonctionnement d'un système de traduction neuronale fr-en construit à l'aide de vastes ressources montre que ni l'une ni l'autre de ces deux hypothèses ne sont totalement correcte : d'une part le contrôle des masques dans la sortie est imparfait, et demande une attention particulière ; d'autre part le système ne corrige en fait qu'un très petit nombre d'erreurs de traduction [7].

2.3 Utilisation de traits linguistiques

Marquage lexical Une forme de contrôle de la traduction consiste à enrichir la présentation des entrées par des caractéristiques linguistiques précalculées par différents prétraitements. De nombreux travaux dans ce sens ont conduit à insérer des pseudo-mots au

7. Qui peuvent toutefois s'avérer très importantes pour la justesse de la traduction du document, leur nombre ne dit donc pas complètement tout ici.

niveau des phrases, pour en contrôler le domaine (Kobus *et al.*, 2017), le style, le niveau de formalité (Niu *et al.*, 2017), voire la langue dans laquelle produire la sortie (Firat *et al.*, 2016). La même démarche peut être opérée au niveau de chaque mot, en combinant la représentation (apprise) du token courant avec une représentation (apprise) des traits linguistiques associés au token. Cette approche est proposée initialement par Sennrich et Haddow (2016), qui combine le plongement du $E(w_t)$ du mot courant w_t avec celui d'un ensemble de K labels (correspondant à des traits typographiques, morphologiques, syntaxiques ou sémantiques) $l_{t,1} \dots l_{t,K}$ pour calculer une représentation enrichie $F(w_t)$ selon par exemple :

$$F(w_t) = [E(w_t), E(l_{t,1}), \dots, E(l_{t,K})] \text{ ou } F(w_t) = E(w_t) + \sum_{k=1}^{K} W_k E(l_{t,k})$$

Cette approche demande une analyse préalable de la source pour l'étiqueter avec les labels qui sont jugés les plus pertinents. Elle présuppose des outils de traitement automatique (étiqueteur morpho-syntaxique, repérage des entités nommées, dépendances, etc) qui peuvent être plus ou moins faciles à obtenir ou délivrer des résultats plus ou moins fiables selon les paires de langues.

Une amélioration globale de la traduction (score BLEU) est observée par Sennrich et Haddow (2016) et confirmée par Post *et al.* (2019) ; en revanche, cette approche ne semble pas significativement meilleure que l'approche de base pour ce qui concerne la traduction des formes extra-lexicales qui sont particulièrement étudiées par ces auteurs.

Une méthode similaire est utilisée dans un contexte de simplification de phrases par Mallinson et Lapata (2019), qui utilisent ces traits à la fois pour identifier les entrées lexicales qui doivent être simplifiées, mais également pour contrôler la structure syntaxique de la phrase en sortie, en injectant dans la source une représentation linéarisée de l'arbre de dépendances syntaxiques.

Prétraduction partielle Han *et al.* (2019) applique et étend cette idée pour intégrer des lexiques bilingues : plutôt qu'utiliser une étiquette, la représentation de chaque mot source est complétée par la représentation d'un mot cible trouvée dans un dictionnaire bilingue. Cette combinaison des deux représentations peut être effectuée de plusieurs manières : somme, concaténation, ou encore combinaison linéaire pondérée (*gating*). Les auteurs étudient une méthode encore plus élémentaire, qui consiste à faire suivre chaque mot source d'un équivalent cible. Dans cette méthode, le choix de l'équivalent dictionnairique à utiliser est critique, ce qui conduit les auteurs à enrichir le pré-traitement par une étape préalable de « désambiguisation sémantique ». En pratique ils extraient des phrases parallèles disponibles à l'apprentissage un dictionnaire bilingue *par alignment automatique* ; au moment du test il s'agira d'associer à la représentation de chaque mot source sa traduction la plus vraisemblable. On note que cette démarche est possible pour la paire de langue considérée (en-zh), pour laquelle la langue cible est morphologiquement très simple ; l'utiliser dans des situations où il faudrait non seulement précalculer le concept source, mais également sa réalisation de source demanderait des extensions de la méthode qui ne sont pas discutées dans cette étude.

Cette stratégie qui consiste à mélanger les langues cible et source par prétraduction, que l'on peut voir comme une version simpliste de la proposition de Niehues *et al.* (2016), qui effectue une prétraduction par une méthode statistique, est poursuivie dans plusieurs travaux récents plus focalisés sur les termes, présentés section 2.4. En plus de sa simplicité, cette démarche présente l'avantage de ne pas perturber le fonctionnement du décodeur neuronal ; au rebours elle fournit peu de garanties sur la sortie produite.

2.4 Prétraduction et alternance codique

Cette stratégie de prétraduction a été adaptée au cadre neuronal dans plusieurs études, avec des variantes mineures. Dinu *et al.* (2019) étudie deux manières d'effectuer cette prétraduction, soit en insérant le terme cible à la suite du terme source, soit en remplaçant le terme source par le terme cible. Dans les deux, cas, les unités source et cible sont identifiés par des labels spécifiques qui font partie des représentations lexicales manipulées par l'encodeur. Ceci implique une représentation conjointe des unités BPE sources et cibles. Ces deux options sont illustrées dans le tableau suivant : L'intérêt de cette approche est

src	All alternates Stellvertreter shall be elected for one term
src + ins	All_0 alternates_1 Stellvertreter_2 shall_0 be_0 elected_0 for_0 one_0 term_0
src + rep	All_0 Stellvertreter_1 shall_0 be_0 elected_0 for_0 one_0 term_0

TABLE 2 – Mélanges source-cible pour la prétraduction de termes. Les indices identifient les différents types d'unités.

qu'elle n'implique aucun changement du décodeur et donc aucun surcoût computationnel de traitement au test. L'objectif principal est de renforcer la propension du décodeur à effectuer des copies et les auteurs n'appliquent donc cette réécriture à l'apprentissage que lorsque le terme et sa traduction apparaissent tous les deux dans le couple de phrases parallèles, possiblement avec des variantes morphologiques repérées par des changements formels mineurs. Des expériences sont menées pour la paire anglais-allemand avec deux listes de traductions (a) un dictionnaire extrait de la wikipedia ; (b) une liste de termes extraits de la terminologie officielle de l'UE [8]. Pour résumer les principales conclusions : des deux méthodes de prétraduction, le remplacement est le plus efficace ; il permet d'obtenir des taux de recopie presque déterministes ($> 90\%$) soit nettement au-dessus de la valeur de base, qui est déjà proche de 77%, en revanche l'impact sur le score BLEU est à peine perceptible [9]

Song *et al.* (2019) étudie une méthode similaire, en proposant au décodeur de traduire des phrases qui contiennent un mélange de langue source et cible. Les mots sources que l'on souhaite contrôler (ici, des noms de personnes, de lieux, d'organisation et de marques, pour les paires de langues zh :en et ru :en) sont simplement remplacés en prétraitement par leur traduction en langue cible, avec pour objectif qu'ils soient directement recopiés

8. IATE : https://iate.europa.eu/

9. Ce qui est attendu puisque le changement n'affecte qu'un mot par phrase dans environ 15% des phrases.

dans la cible. Pour maximiser les chances que cette recopie soit effectuée, les auteurs entrainent leur système avec des phrases « mêlant » de manière contrôlée source et cible. Pour engendrer le corpus d'apprentissage, on utilise donc un alignement automatique source-cible qui permet de substituer aléatoirement des segments sources par des segments cibles ; ces segments cibles étant présents à l'identique côté cible, le décodeur est donc incité à apprendre à effectuer ces copies. Pour accentuer la production de copies, les auteurs proposent également d'utiliser un réseau pointeur (*pointer network*) (Gu *et al.*, 2016) (voir section 4.1), ce qui un effet très sensible sur les performances, comme déjà noté par Pham *et al.* (2018). Comme dans l'étude de Dinu *et al.* (2019), les représentations sources et cibles sont nécessairement partagées, au moins du côté de l'encodeur, ce qui est un autre facteur facilitant la traduction. Le taux de copie calculé sur des tâches de traduction anglais-russe et anglais chinois est de l'ordre de 90%. Par rapport à l'approche précédente, ce système est entrainé avec de nombreux exemples d'alternance codique, ce qui lui permet de maintenir des performances de traduction identique même lorsque plusieurs traductions sont effectuées.

3 Contraintes de décodage en cible

Faute de disposer d'un alignement entre source et cible, certains travaux proposent d'intervenir dans le processus de décodage du système de traduction et d'appliquer les contraintes uniquement dans la cible sur les hypothèses générées au cours de la recherche en faisceau (*beam search*). L'objectif est donc une moins ambitieux et vise à simplement à assurer que, dès lors que certains termes sont repérés dans la source, alors certaines unités devront apparaitre quelque part (au moins une fois, ou exactement une fois) dans la cible produite. Des éléments de compréhension des algorithmes impliqués dans ce processus sont rappelés à la section 1.1.2.

3.1 Décodage contraint

Nous présentons ici une famille de méthodes qui proposent d'intervenir dans le processus de décodage du système de traduction et d'appliquer des contraintes sur les hypothèses générées au cours de la recherche en faisceau.

L'approche de Hokamp et Liu (2017) contraint le décodage en forçant la présence des contraintes cibles dans l'hypothèse générée. La tâche du décodeur consiste alors à insérer ces mots aux meilleurs endroits possibles dans la phrase cible et aucun lien à des parties de la phrase source n'est explicitement modélisé. Concrètement, l'algorithme de décodage maintient C ensembles de préfixes actifs à chaque instant, chaque ensemble $B_{t,i}, i = 0 \dots C-1$ stockant les préfixes pour lesquels i contraintes ont déjà été satisfaites. Le développement de la k$^{\text{ème}}$ hypothèse de la i$^{\text{eme}}$ pile $\mathbf{e}_{<t,k,i}$ conduit alors à une nouvelle hypothèse $\mathbf{e}_{<t+1,k',i'}$, qui soit satisfait une contrainte supplémentaire ($i' = i + 1$), soit ne satisfait pas de nouvelle contrainte ($i' = i$).

Cette approche complexifie la recherche en faisceau en le multipliant le nombre d'hypothèses à maintenir par un facteur qui croit linéairement avec le nombre de contraintes),

ce qui, en pratique, ralentit significativement le décodage. Elle complique plus généralement l'implémentation car le nombre de contraintes (donc d'ensembles actifs) varie de phrase en phrase et interdit d'optimiser le décodage par block sur GPU. Elle pose enfin la question de la satisfaction partielle des contraintes (en présence d'unités polylexicale ou sous-lexicales), du décompte du nombre de fois où une contrainte est satisfaite et impliquera également dans les faits un certain nombre d'opérations ancillaires pour l'algorithme de décodage. Comme pour l'algorithme de décodage de Moses, chaque pile contient des hypothèses qui satisfont des contraintes différentes et dont les scores ne sont pas forcément directement comparables.

Post et Vilar (2018) propose une variante de cette approche qui permet de retrouver un temps de décodage similaire à celui de la configuration de base (sans contrainte) [10]. Le principe général est de renoncer à maintenir B hypothèses pour chaque nombre de contraintes possiblement satisfaites, mais au contraire de fixer B, et de répartir les positions au sein de B_t parmi les C types d'hypothèses (chacune ayant par exemple $\lfloor \frac{B}{C} \rfloor$ positions dans B_t). À chaque pas de temps on n'étend donc que B préfixes, dont les continuations seront évaluées selon le nombre de contraintes qu'elle satisfont. On note que pour que cette approche soit effective, il est préférable de choisir $B > C$.

3.2 Limites du décodage contraint

Ces méthodes sont effectives, à un coût computationnel qui reste contrôlé, mais impliquent un usage strict des contraintes lexicales. Il peut en effet arriver que le décodeur attribue un score extrêmement bas à l'un des mots impliqués dans une contrainte, ce qui aura pour effet de contrarier la suite du processus de décodage, puisque le système se retrouve dans une zone peu explorée de l'espace de recherche. La suite de la traduction est alors souvent mauvaise, voire absente.

Par ailleurs, les contraintes sont appliquées strictement et aucun mécanisme d'adaptation au contexte n'est mis en place. Ainsi, si le lexique d'où sont tirées les contraintes ne comprend que des verbes à l'infinitif, il n'est pas possible de laisser le système de traduction les conjuguer. Cette particularité restreint l'application de cette méthode à des mots invariables, comme des noms propres ou des acronymes et exclut les langues dites « à morphologie riche », pour lesquelles la très grande majorité des formes se fléchit. Pour ces langues, le mécanisme qui applique une contrainte devrait disposer de connaissances morpho-lexicales afin d'adapter la cible à différents contextes sources et cibles.

Ces contraintes doivent en somme être assouplies pour une meilleure adaptation aux modèles neuronaux. Cette adaptation peut passer par une meilleure intégration au processus de décodage. Le décodeur devrait par exemple avoir la possibilité de rejeter une contrainte qu'il juge mauvaise et privilégier une variante ou un synonyme. Cela permettrait d'éviter toutes sortes de perturbations qui peuvent conduire à des sorties défectueuses.

10. En pratique, cette augmentation du décodage entraîne un surcoût computationnel non négligeable à case des opérations auxiliaires à effectuer pour distribuer les hypothèses dans les différentes piles.

3.3 Affaiblissement des contraintes et modèles de cache

Les travaux de Feng *et al.* (2017); Wang *et al.* (2017b), puis récemment de Li *et al.* (2019a) s'inscrivent dans une ligne de recherche qui vise à améliorer la prise en compte des contraintes en transformant les traitements en cible. Ici, les contraintes sont traitées comme des ressources complémentaires au moment du calcul des mots qui sont proposés par le décodeur.

Feng *et al.* (2017) proposent de réexprimer les contraintes sous une forme adaptée au décodage : à partir d'un réservoir global de contraintes (toutes les associations préenregistrées), l'analyse de la source permet d'identifier les contraintes actives localement, représentées par l'association de la représentation contextuelle d'un mot source d'intérêt avec la sortie désirée. Un module d'attention dédié entre alors en jeu au moment du décodage pour déduire de cette mémoire locale la probabilité à postériori de chaque mot cible apparaissant dans une contrainte, qui est finalement interpolée avec la distribution calculée par le décodeur. Plusieurs manières de calculer cette attention sont comparés, l'approche la plus efficace consistant à utiliser simultanément comme « clé » l'état interne du décodeur et le dernier mot traduit. Les mots qui ne sont concernés par aucune contrainte restent traduits seulement par le modèle de traduction. Cette approche est conceptuellement similaire à la fusion du modèle neuronal avec un modèle de cache, qui permettrait de renforcer la probabilité unigramme des mots cibles impliqués par les contraintes. Ce modèle est complété pour pouvoir aussi traiter les mots hors-vocabulaire, qu'ils apparaissent en source ou en cible.

Dans (Wang *et al.*, 2017a,b), la méthode repose sur une combinaison entre traduction neuronale et traduction statistique au moment du décodage : à chaque pas de temps, le décodeur à la possibilité de traduire conventionnellement, ou de consulter une mini-table de segments qui est recalculée à partir du préfixe courant – et peut intégrer également des éléments lexicaux définissant un ensemble de contraintes. À l'entraînement, les paramètres de la porte (un réseau multicouche) qui » contrôle « cette décision sont appris avec les autres paramètres du modèle. Contrairement aux approches de Dahlmann *et al.* (2017); Zhao *et al.* (2018b) discutées ci-dessous, le module d'alignement n'est pas sollicité dans ce modèle, et les hypothèses du modèle à base de segments sont construites indépendamment de celles du décodeur neuronal.

La contribution plus récente de Li *et al.* (2019a) réexprime cette alternative d'une manière un peu plus souple. Comme les autres travaux de cette section, les contraintes sont intégrées comme un contexte supplémentaire conduisant à générer le prochain mot selon $p(e_t|e_{<t}, f, C)$, où C représente l'ensemble des contraintes. Conceptuellement, cette méthode est très similaire à l'idée de fusion « profonde » du système de traduction avec un modèle de langue cible, discutée à la section 1.1.3. Un avantage est que la recherche de la meilleure solution a ici exactement la même complexité que le décodage standard. Plusieurs manières d'implanter concrètement cette règle de décision dans une architecture Transformer sont considérées, et des expérimentations sont conduites sur deux couples de langues. Les performances sont comparables aux résultats de Post et Vilar (2018), avec un avantage supplémentaire que les contraintes peuvent contenir du bruit ou des erreurs. Cette approche répond bien à quelques unes des critiques adressées aux méthodes

à base de masque, mais à l'inverse, elle fournit des garanties relativement faibles quant à l'utilisation de contraintes, puisque le choix d'appliquer ou pas la contrainte est soumis à la décision probabiliste du décodeur. La partie expérimentale, qui utilise des contraintes non-réalistes, ne permet pas de mesurer à quel point les contraintes sont effectivement appliquées.

3.4 Les contraintes comme régularisation

La proposition de Zhang *et al.* (2017a) se distingue des précédentes dans sa manière d'intégrer des contraintes à travers un modèle auxiliaire, qui s'appuie sur le concept de régularisation postérieure de Ganchev *et al.* (2010). Le principe général est de s'assurer que la distribution conditionnelle $p_{\boldsymbol{\theta}}(\mathbf{e}|\mathbf{f})$ reste proche d'une distribution à priori $q_{\boldsymbol{\gamma}}(\mathbf{e}|\mathbf{f})$ qui intègre les contraintes dans un modèle log-linéaire selon :

$$q_{\boldsymbol{\gamma}}(\mathbf{e}|\mathbf{f}) = \frac{1}{Z_{\boldsymbol{\gamma}}(\mathbf{f})} \exp(\boldsymbol{\gamma}^T F(\mathbf{f}, \mathbf{e})), \tag{3}$$

avec $F()$ le vecteur des contraintes. $F()$ peut ainsi prendre en compte le fait que $\mathbf{f}$ et $\mathbf{e}$ contiennent des mots ou des phrases présents dans des dictionnaires, intégrer des contraintes de couverture « molles » (à travers l'attention), contrôler les rapports de longueur, etc. D'une certaine manière, on peut comprendre cette méthode comme une tentative de régulariser un modèle neuronal par un modèle statistique log-linéaire, comme le proposent Ren *et al.* (2019) dans un contexte de TA neuronale *non supervisée*. Un des points forts de cette méthode est son caractère générique, qui permet de modéliser des contraintes de nature très variées.

La fonction objectif à optimiser lors de l'apprentissage inclut alors un terme de régularisation et prend la forme suivante :

$$\sum_i \lambda_1 \log p_{\boldsymbol{\theta}}(\mathbf{e}^{(i)}|\mathbf{f}^{(i)}) - \lambda_2 \,\mathrm{KL}(p_{\boldsymbol{\theta}}(\mathbf{e}|\mathbf{f}^{(i)})||q_{\boldsymbol{\gamma}}(\mathbf{e}|\mathbf{f}^{(i)})),$$

où KL est la divergence de Kullback-Leibler entre les deux distributions. Si la fonction objectif ainsi définie est convexe et complètement dérivable, elle est toutefois impossible à calculer exactement à cause de l'espérance (sur toutes les phrases cibles) impliquée dans le calcul de la divergence KL. Comme il est d'usage, les auteurs proposent de s'appuyer sur des méthodes d'échantillonnage à l'apprentissage. Au décodage d'une nouvelle instance, il s'agira de trouver une traduction $\mathbf{e}$ qui soit à la fois probable pour $p_{\boldsymbol{\theta}}()$ et pour $q_{\boldsymbol{\gamma}}()$, un problème que les auteurs traitent par recherche locale à partir d'une solution initiale. L'amélioration du score BLEU obtenu (sur des données chinois-anglais) reste toutefois faible et les auteurs ne précisent pas à quel point les contraintes sont effectivement satisfaites dans ce modèle, qui ne peut toutefois fournir que des garanties faibles. Une autre manière d'apprécier cette approche consiste à réaliser qu'en forçant l'apprentissage d'un modèle ressemblant d'un système de TA statistique, il est à craindre d'altérer globalement les performances des systèmes de TAN, qui sont en général meilleurs que leurs équivalents statistiques.

Si cette approche a été peu suivie pour la traduction neuronale, on notera qu'elle trouve un écho dans le travail de Zhao *et al.* (2018a), qui propose une idée similaire pour imposer l'utilisation de paraphrases de la *Paraphrase DataBase* (PPDB) (Ganitkevitch *et al.*, 2013) dans un contexte de simplification automatique de phrases.

4 Utilisation de l'alignement lors du décodage

Dans cette section, nous présentons diverses études qui visent à informer le décodage neuronal par des informations d'alignement. Ceci pouvant être mis en œuvre soit en exploitant le dispositif d'attention (voir la section 1.1) qui est disponible par défaut, soit en améliorant le mécanisme d'attention pour le rendre équivalent à un alignement.

4.1 L'attention comme alignement

4.1.1 Lexiques dynamiques et recopie

La possibilité de recopier directement certains mots sources dans la cible est essentielle pour traiter correctement les extra-lexicaux (cf. la discussion de la section 1.2). Gu *et al.* (2016) proposent une architecture qui rend explicite cette possibilité, et qui consiste pour l'essentiel à combiner au moment de la génération de chaque mot source les sorties de deux modèles :

— le modèle standard du décodeur neuronal qui associe à chaque mot e du *vocabulaire cible* une probabilité $p(e|\mathbf{e}_{<t}, \mathbf{f})$; il est possible via ce modèle de produire un token correspondant à un mot inconnu ;

— un modèle de recopie qui associe à chaque mot f de la *phrase source* une probabilité de recopie $p(f|\mathbf{e}_{<t}, \mathbf{f})$

On notera que ces deux situations ne sont pas exclusives l'une de l'autre, les vocabulaires source et cible pouvant avoir une intersection non vide.

Les paramètres des ces deux modèles sont combinés d'une manière qui n'est pas complètement équivalente au mélange standard qu'opèrerait par exemple une fusion superficielle de modèles de langue, puisque la formulation précise du modèle donne lieu aux trois cas suivants, qui *partagent le même facteur de normalisation*[11] :

$$p(y|\mathbf{e}_{<t}, \mathbf{f}) \propto \begin{cases} \sum_{i,y=\mathbf{f}_i} \exp \phi_c(y, \mathbf{e}_{<t}, \mathbf{f}) & \text{si } y \in \mathbf{f} \textit{ absent} \text{ de } V_\mathbf{e} \\ \sum_{i,y=\mathbf{f}_i} \exp \phi_c(y, \mathbf{e}_{<t}, \mathbf{f}) + \exp \phi_g(y, \mathbf{e}_{<t}, \mathbf{f}) & \text{si } y \in \mathbf{f}, \textit{présent} \text{ dans } V_\mathbf{e} \\ \exp \phi_g(y, \mathbf{e}_{<t}, \mathbf{f}) & \text{sinon (inclut } \texttt{<unk>}) \end{cases}$$

Dans cette approche, le mécanisme d'alignement standard impliqué dans le calcul de $\phi_g(y, \mathbf{e}_{<t}, \mathbf{f})$ est complété par un second mécanisme plus précis, destiné à encourager les séquences de copies, qui sont fréquentes, en particulier, pour les noms propres.

Une architecture très similaire est présentée dans (Gulcehre *et al.*, 2016), qui mélange également deux modèles de génération (de manière superficielle) : la génération neuronale

11. Les sommes pour les mots sources prennent en compte la possibilité qu'un mot apparaisse à plusieurs positions.

standard et la génération de la position source à recopier. Le mélange est entrainée de manière supervisée en recalculant de manière heuristique pour chaque mot s'il a été engendré par l'un ou l'autre des composants. Zhou *et al.* (2018) en proposent une version capable de prendre en compte la recopie de segments, ce qui demande d'identifier, en cas de recopie, les indices de début et la fin de l'empan qui est recopié ; diverses autres complications, par exemple de l'algorithme de recherche, sont également présentées. Cette technique n'est évaluée que pour des applications de résumé par extraction.

Il en va pour la recopie comme pour le traitement des mots inconnus : la généralisation des modèles fondés sur des unités sous-lexicales (partagées entre les deux langues) tend à faciliter la recopie sans qu'il soit besoin d'ajouter un composant supplémentaire. C'est l'une des conclusions de l'étude de Knowles et Koehn (2018), qui analyse sur le couple de langues anglais-allemand les unités qui sont – ou non – recopiées dans un système standard.

4.1.2 Décoder avec un dictionnaire probabiliste

Les travaux de Arthur *et al.* (2016) s'intéressent à l'utilisation de dictionnaires, éventuellement probabilistes, comme ressources complémentaire qui viennent mieux informer la décision lors du décodage. Contrairement aux travaux de la section 3.3, le score d'attention est utilisé comme substitut de l'alignement pour repérer à chaque instant les contraintes les plus pertinentes. Formellement, la première étape du traitement construit un sous-dictionnaire probabiliste associant à chaque mot source f le vecteur L_f de dimension V_e, avec $L_f[e] = p(e|f)$, où les probabilités lexicales sont préentrainées avec des modèles probabilistes conventionnels. Le même mot source apparaissant à des positions différentes donnera lieu au même vecteur. En multipliant la matrice formée des colonnes $[L_{\mathbf{f}_1} \dots M_{\mathbf{f}_J}]$ par le vecteur d'attention $\boldsymbol{\alpha}_i$, on obtient à chaque instant une probabilité lexicale qui vient compléter le score calculé par le décodeur. Deux manières de combiner ces informations sont étudiées : soit en intégrant cette probabilité lexicale comme un biais dans la couche de sortie, soit en interpolant linéairement les deux probabilités (à la manière d'une fusion superficielle, puisque l'on combine en fait deux modèles de langue cible). Nguyen et Chiang (2018) revisitent ce travail, en modifiant la manière dont le modèle local des mots sources est estimé : plutôt que d'utiliser un modèle discret préentrainé, les auteurs calculent ce nouveau contexte comme une fonction de la moyenne (pondérée par l'attention) des plongements sources. Dans leurs expériences avec plusieurs paires de langues peu dotées, leur approche s'avère meilleure (et moins coûteuse) que celle de Arthur *et al.* (2016).

Cette approche est étendue par Zhao *et al.* (2018b), qui remplacent le dictionnaire de Arthur *et al.* (2016) par la table des segments d'un système statistique. L'étape de préparation repère, pour chaque mot source, les phrases potentiellement intéressantes ; en utilisant le calcul de l'attention à chaque étape de décodage, on peut réévaluer la pertinence de chacune d'entre elles en fonction du préfixe courant qu'elles pourraient développer ; la couche de sortie combinera ces prédictions avec celles proposées par le décodeur. Des améliorations du score BLEU sont observées pour les paires anglais-japonais et anglais-chinois, pour une méthode qui s'avère finalement très proche de (Dahlmann

et al., 2017) détaillée ci-après.

4.1.3 Guider le décodage

Chatterjee *et al.* (2017) appliquent au décodage des contraintes sous la forme de paires (source→ cible). Lors de la génération d'une hypothèse e_i, la méthode s'appuie sur le mécanisme d'attention pour identifier le mot ou segment source en cours de traduction f_j ; si ce segment est associé à une contrainte $(f_j \rightarrow e'_i)$, alors le mot cible proposé e_i par le système est directement substitué par application de la contrainte. Cette vérification est appliquée à chacune des hypothèses actives dans le faisceau. Afin d'améliorer le bon positionnement de la réécriture dans la phrase cible, l'algorithme de décodage est augmenté d'un mécanisme de regard avant (*look-ahead*) : ceci afin d'éviter d'appliquer à l'instant i une contrainte que le décodeur aurait « spontanément » appliqué à l'instant $i + k$. Pour être complet, ajoutons que les auteurs complètent ce dispositif afin de traiter (par recopie) également les mots hors-vocabulaires, une modification qui s'avère décisive pour traiter les extra-lexicaux et les termes inconnus et améliorer les performances par rapport au modèle de base. Utiliser l'attention comme alignement pose plusieurs problèmes.

D'une part, elle part de l'hypothèse que le mécanisme d'attention s'apparente à un alignement, ce qui n'est pas exactement le cas (Koehn et Knowles, 2017; Ghader et Monz, 2017). Contrairement aux alignements déterministes qui sont binaires, les poids de l'attention sont des valeurs continues. Pour savoir si un état donné du décodeur s'apprête à traduire un mot source particulier, on se fonde sur le poids qui lui est attribué par l'attention. Or il est possible que ce même mot source se verra attribuer un poids encore plus élevé dans un état postérieur du décodeur. Pour répondre à ce problème, le décodeur doit explorer des états ultérieurs afin de s'assurer que le mot source sujet à la contrainte ne sera pas (re)traduit plus tard. Ces explorations ralentissent considérablement le décodage, le rendant irréaliste dans un contexte industriel. Enfin, cette méthode repose sur des systèmes comprenant un unique mécanisme d'attention (Bahdanau *et al.*, 2014), mais demanderaient une réévaluation du fait du développement de modèles qui en contiennent plusieurs, comme le modèle Transformer de Vaswani *et al.* (2017a).

4.1.4 Décodage et contraintes rationnelles

Une seconde limitation de l'approche de Chatterjee *et al.* (2017) est l'absence d'un contrôle explicite du nombre de contraintes qui sont satisfaites lors du décodage, même si cette limitation ne semble pas poser de problème particulier dans leurs expérimentations. Cette limitation est levée par Hasler *et al.* (2018), qui réexpriment le problème sous la forme d'un décodage sous contraintes rationnelles, représentables par un automate fini. Cette formalisation est déjà présente dans (Anderson *et al.*, 2017) qui s'intéresse à la génération de légendes pour des images. Dans sa version la plus naïve, l'ensemble des contraintes est représentée comme un automate fini et chaque état de l'automate est associé à un ensemble d'hypothèses actives pour la recherche en faisceau, ce qui assure l'homogénéité des hypothèses, avec un coût qui devient vite prohibitif si les contraintes sont purement lexicales. En effet, l'automate représentant la satisfaction de C contraintes

(dans un ordre quelconque) possède 2^C états, un état pour chaque sous-ensemble possible de contraintes. Pour limiter le coût du décodage, Hasler *et al.* (2018) proposent d'utiliser à chaque pas de temps t le mécanisme d'attention pour sélectionner, hypothèse par hypothèse, les contraintes qui sont actives pour le segment courant. De cette manière, on espère éviter que tous les états de l'automate des contraintes ne soient ni construits ni visités, puisqu'à chaque instant chaque hypothèse ne conduit qu'à explorer au plus un nouvel état de l'automate des contraintes. D'autres techniques pour garantir la correction de la sortie, utilisant notamment des contraintes de couverture, sont également implantées. D'une certaine manière, ce travail réalise une hybridation entre les méthodes à base de contraintes cibles (section 3) et les méthodes à base d'alignement / attention (section 4).

4.1.5 Vers des systèmes hybrides : le retour des « segments »

La proposition de Dahlmann *et al.* (2017) est représentative d'une série de travaux (Tang *et al.*, 2016; Wang *et al.*, 2017b; Zhang *et al.*, 2017b) visant à construire des modèles hybrides neuronal / statistique, susceptibles de tirer partie de la table de traduction d'un modèle à base de segments (Koehn, 2010). Cette table contient non seulement les couples (de taille variable) source-cible jouant le rôle de contraintes, mais également les scores des modèles à base de segment (probabilités lexicales bidirectionnelles, modèle de langue). En conséquence, la proposition de Dahlmann *et al.* (2017) consiste à modifier le décodage en faisceau de manière à prendre en compte les scores d'attention et les utiliser pour repérer et réévaluer, à chaque pas de temps, les entrées de la table des segments qui pourront aider à développer les hypothèses de traduction. En maintenant à jour un historique des vecteurs d'attention passés, le décodeur évite d'appliquer plusieurs fois les mêmes contraintes et reconstruit implicitement une vision de la couverture source. Par rapport aux méthodes qui se focalisent uniquement sur la traduction de termes, cette approche implique un recours bien plus massif aux ressources lexicales durant la recherche. Dans cette approche hybride, la division du travail entre les différents modèles est inégale, puisque le modèle neuronal calcule l'attention et est responsable de l'ordre dans lesquels les mots sont produits ; le modèle statistique se contentant de fournir des scores auxiliaires. On peut penser que dans ce modèle, le décodeur neuronal est peu perturbé dans son fonctionnement, mais qu'à l'inverse les garanties de voir des entrées de la table des segments effectivement utilisées sont faibles. Les expérimentations montrent toutefois que l'effet est très positif pour la traduction des mots (ou expressions polylexicales) rares qui peuvent être sur-segmentées en unités courtes, ce qui peut conduire à des métraductions massives.

Parmi les travaux de cette famille, (Park et Tsvetkov, 2019) se démarque par la manière dont il manipule toutes les représentations lexicales – y compris celles impliquées par les contraintes – dans des espaces continus. Ces auteurs présentent ainsi un système qui engendre non pas des séquences de mots, mais des séquences de représentations / plongements lexicaux (représentant des unités mono- ou poly-lexicales). Ce système présente deux innovations majeures : un composant neuronal qui prédit la fertilité de chaque mot source ; une stratégie pour apprendre sur des corpus monolingues des plongements

lexicaux pour des expressions ou termes complexes (restreints ici à associer un unique mot source à plusieurs mots cibles).

L'apprentissage se déroule avec des segmentations de référence : à chaque pas de temps on entraîne le modèle à produire le bon mot (ou terme) avec un critère d'apprentissage standard. Une fois cette étape réalisée, on entraine le modèle de fertilité. Au décodage (probablement glouton, les auteurs ne le précisent pas), on choisit d'abord la fertilité, puis le plongement cible. Cette approche a le mérite de conserver sa simplicité au décodage, puisque les mots et termes sont traités de manière identique et comptent pour une unité de sortie. Deux écueils sont ainsi évités : la complexité d'un décodage avec des unités de taille variable ; le calcul d'une couche de *softmax* qui devrait sélectionner dans un vocabulaire cible mêlant termes et mots simples. Les résultats expérimentaux obtenus pour la paire de langue allemand-anglais sont mitigées : les variations du score BLEU sont modestes, la méthode proposée se montrant toutefois à son avantage lorsque l'on se restreint à l'évaluation de la traduction des unités polylexicales.

4.2 Traduire avec un alignement explicite

Les valeurs calculées par le module d'attention ne fournissent qu'une approximation de l'alignement source cible. Nous présentons dans cette section divers travaux qui intègrent un alignement dans la traduction neuronale dans le but exprès de pouvoir injecter des contraintes lexicales. D'autres travaux s'intéressent à l'amélioration de l'attention pour améliorer la traduction en général, en introduisant des biais (Cohn *et al.*, 2016) ou en supervisant l'apprentissage du module d'attention par des alignements automatiques (Mi *et al.*, 2016b; Liu *et al.*, 2016), ou encore (Garg *et al.*, 2019) pour une adaptation au modèle Transformer. Il existe plusieurs manières de parvenir à ce but : comme dans les études précédentes, en rajoutant un terme à la fonction objectif qui assure que les vecteurs d'attention diffèrent peu des alignements de références – on se rapproche alors du cadre de l'apprentissage multi-tâche ; soit en intégrant l'alignement comme une variable de plein droit du modèle.

4.2.1 Apprentissage multi-tâche

(Chen *et al.*, 2016) s'inscrit dans la première de ces deux alternatives, et propose plusieurs extensions à l'architecture d'un traducteur neuronal : d'une part, la supervision des matrices d'alignement par un alignement de référence à l'apprentissage ; d'autre par l'injection d'une variable décrivant le thème du document dans le décodeur. Grâce à de meilleurs alignements, ces auteurs proposent de traduire les mots inconnus (ou des extra-lexicaux) en combinant deux techniques : (a) en pré-traitement, le masquage (§ 2.2) des unités sources à remplacer ; (b) au décodage, le remplacement du masque cible (une fois généré) par le mot source qu'il traduit. Cette seconde étape bénéficie de la supervision du modèle d'attention. La combinaison de ces deux techniques leur permet d'obtenir des améliorations substantielles dans une application de traduction automatique pour le domaine du commerce en ligne.

À l'instar de (Garg *et al.*, 2019), la démarche de Song *et al.* (2020) (schématisée sur

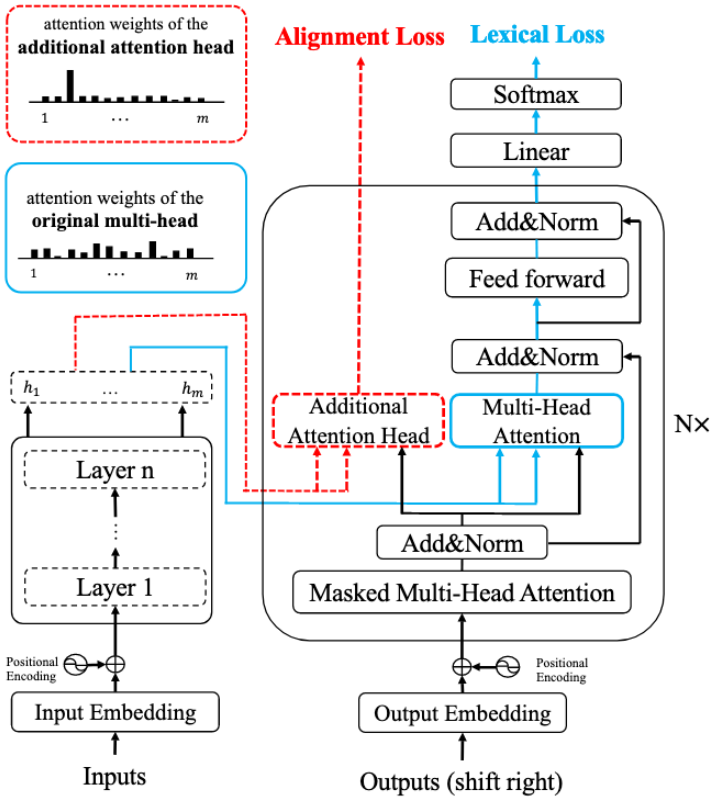

FIGURE 4 – L'architecture de Song *et al.* (2020). Figure reproduite d'après l'article cité.

la figure 4) vise à superviser le module d'attention d'un modèle Transformer en utilisant des alignements de « référence » [12].

Elle s'en démarque toutefois en lui allouant à cette tâche une « tête » spécifique qui calcule un ensemble de variables distinct des variables d'attention, et qui n'interviennent pas dans le calcul de la traduction. En conséquence, le système est entrainé à réaliser conjointement deux prédictions : d'une part l'alignement, d'autre part le mot cible. La qualité ces deux prédictions peut être évaluée séparément : en termes d'AER (*Alignment Error Rate*) pour le premier, avec des améliorations massives par rapport à un modèle sans supervision ; en termes de score BLEU pour le second, avec des améliorations pour plusieurs paires de langues. De manière complémentaire, les auteurs mesurent la qualité des alignements selon un protocole déjà utilisé par (Hasler *et al.*, 2018; Post et Vilar, 2018) : une fois le jeu de test aligné avec la référence, un dictionnaire « idéalisé » est extrait et est utilisé pour choisir le prochain mot chaque fois que l'alignement vers la source identifie un mot du dictionnaire. Disposer des alignements corrects permet de reproduire les bons choix de traduction ; à l'inverse, les erreurs d'alignement conduiront automatiquement à des erreurs de traduction.

12. Qui sont en fait calculés par les outils d'alignement statistique.

4.2.2 Le retour des alignements

La proposition de Alkhouli et Ney (2017) est substantiellement différente et intègre explicitement une représentation de l'alignement qui vient complèter et enrichir le module d'attention [13]. À l'apprentissage, on observe à chaque instant le mot cible et son alignement ; au décodage, on devra prédire successivement ces deux variables et donc choisir chaque mot avec un modèle très semblable aux modèles IBM historiques de Brown *et al.* (1990) selon [14] :

$$p_\theta(\mathbf{e}_t|\mathbf{e}_{<t},\mathbf{f}) = \sum_k p_\theta(a_t = k|\mathbf{e}_{<t},\mathbf{f}) * p_\theta(\mathbf{e}_t|a_t,\mathbf{e}_{<t},\mathbf{f})$$
$$\max_k p_\theta(a_t = k|\mathbf{e}_{<t},\mathbf{f}) * p_\theta(\mathbf{e}_t|a_t,\mathbf{e}_{<t},\mathbf{f})$$

L'intérêt de cette seconde approche est de pouvoir apprendre et exploiter un modèle d'alignement (récurrent dans cette étude) qui vient complèter le modèle de traduction (et son composant attentionnel). Les auteurs montrent ainsi des améliorations de la traduction, ainsi que des alignements calculés par le système. L'idée d'un apprentissage conjoint de l'alignement et de la traduction est exploitée ci-dessus dans divers travaux.

Cette stratégie est poursuivie dans (Alkhouli *et al.*, 2018), qui la transpose au modèle des Transformers. Conformément à la philosophie de ce modèle, la variable d'alignement est également calculée par une modèle auto-attentif (plutôt que récurrent), et permet de calculer la sortie d'une tête additionnelle qui dérive de cet alignement ; combinée avec la sortie des autres têtes, ce composant vient informer le choix du prochain mot cible. En plus d'améliorer la traduction et/ou la vitesse de décodage, cette méthode est évaluée par sa capacité à prédire correctement le prochain mot traduit *lorsque le décodage est contraint par un dictionnaire*, selon le même protocole (artificiel) que (Song *et al.*, 2020) décrit ci-dessus.

5 Bilans et perspectives

5.1 Une vue d'ensemble

La prise en compte de ressources ou contraintes lexicales durant le décodage prend des formes variées, qui répondent de manière plus ou moins stricte aux attentes que l'on peut porter à ces méthodes, mais qui impliquent également des interventions plus ou moins coûteuses à divers moments du processus d'apprentissage ou d'exploitation d'un système de traduction neuronale. Une vision résumée de ces diverses méthodes est donnée dans le tableau 3.

La figure 5 donne une autre représentation d'ensemble de ces techniques, qui permet de mieux visualiser les mécanismes de contrôle et leur position dans le flux de traitement des

13. Comme le notent en effet également Kim *et al.* (2017), l'alignement dans le modèle standard n'est pas une variable aléatoire, mais résulte d'un calcul déterministe qui implique les états sources et cibles du décodeur.

14. En pratique les deux modèles sont pondérés de manière à équilibrer leur contribution à la décision finale

Méthode	Impact	Types	Max	Force	Place	Forme
Adaptation au domaine (§ 1.1.3)	A	tous	N	-	-	-
Dictionnaires = corpus (§ 2.1)	-	tous	N	-	-	-
Rétro-traduction (§ 2.1)	-	tous	N	-	-	-
Masquage / démasquage (§ 2.2)	P	mots	O	+	+	±
Traits linguistiques (§ 2.3)	P	tous	N	-	-	-
Alternance codique (§ 2.3)	P	termes	O	+	±	±
Recopie mots sources (§ 4.1)	(A)+D	mots	O	±	±	–
Contraintes en cible (§ 3.1)	D	tous	O	++	-	±
Fusion de LM (§ 3.3)	(A)+D	tous	N	-	±	+
Régularisation postérieure (§ 3.4)	A	termes	N	-	-	±
Attention + remplacement (§ 4.1)	D	tous	N	+	+	±
Alignement + remplacement (§ 4.2)	A+D	tous	tous	++	+	±

TABLE 3 – Intégrer des contraintes dictionnairiques - un bilan. La colonne *Impact* indique si la méthode affecte l'**A**pprentissage, le **D**écodage, ou bien les **P**ré/Posttraitements. *Type* indique quels types d'unités peuvent être concernés par la méthode, et *Max* si le nombre de contraintes applicables / phrase est limité. Les trois colonnes suivantes évaluent respectivement le contrôle que chaque méthode fournit vis-à-vis de la présence de la contrainte cible, le contrôle de la position à laquelle la contrainte s'exerce, enfin le contrôle de la forme du segment cible. ± signifie que ce contrôle peut-être laissé au décodeur.

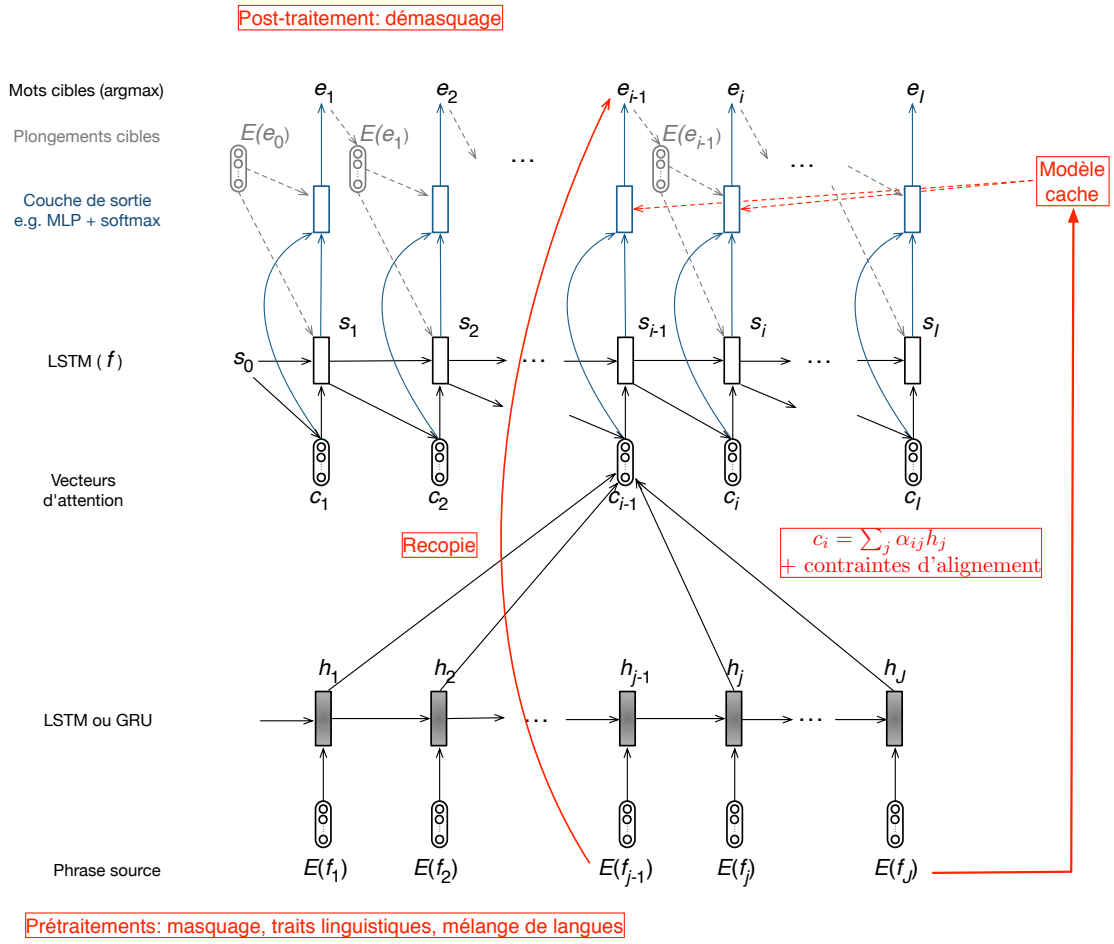

FIGURE 5 – Injection de contraintes dans une architecture neuronale.

données. Elle permet également d'identifier un certain nombre d'angles morts, dont certains pourront faire l'objet de développements ultérieurs et qui sont discutés ci-dessous.

5.2 Angles morts

5.2.1 Méthodes

Une grande variété de principes méthodologiques motivent les approches recensées dans ce rapport. Deux grandes manières de traiter les données dictionnairiques se dégagent toutefois : soit conserver leur forme initiale d'une réécriture plus ou moins inconditionnelle ; soit transformer l'ensemble des contraintes et le voir comme un modèle statistique supplémentaire, qui est combiné au modèle neuronal pendant l'apprentissage ou durant la recherche de la traduction optimale. Une autre façon d'organiser ce panaroma consiste à distinguer deux manières de traiter les contraintes : l'une (classique) qui les exprime dans

des représentations discrètes et exige de revenir aux unités lexicales (cibles ou sources) manipulées par le système de traduction ; l'autre (plus conforme à l'esprit des méthodes neuronales) consiste à réexprimer les contraintes sous la forme de représentations continues.

Plusieurs familles de méthodes sont absentes de ce panorama. D'une part l'utilisation de critères d'apprentissage multi-tâche (Caruana, 1997)qui pourraient inciter l'encodeur et le décodeur à être plus sensibles à la présence de termes ou d'entités à contrôler. Cette stratégie est par exemple utilisée dans un contexte d'adaptation au domaine par Britz *et al.* (2017), qui ajoute à la fonction de perte un terme de prédiction du domaine (en source ou en cible) - la même démarche pourrait être utilisée pour traiter les entrées dictionnairiques, termes ou expressions multi-mots.

D'autre part, l'utilisation de techniques adaptées de sélection et de préentrainement des entrées dictionnairiques ou terminologiques. Si les ressources parallèles sont souvent rares en domaine de spécialité, les ressources monolingues sont probablement plus simple à collecter : adapter les unités et leur représentation sur des corpus monolingues comparables semble de nature à améliorer la traduction de lexiques techniques, comme elles ont pu aider à améliorer la traduction dans des domaines généraux (Conneau et Lample, 2019; Edunov *et al.*, 2019).

On note enfin une absence quasi complète de travaux visant à améliorer *l'explicabilité* des systèmes de traduction, alors que ce sujet est un des plus brûlants qui soit dans le domaine de l'IA en général. Mentionnons toutefois le travail de (Stahlberg *et al.*, 2018b), qui recycle le modèle de séquence d'opérations (*operation sequence model*) développé pour la traduction statistique (Durrani *et al.*, 2015) à des fins de production d'un alignement source-cible qui servira d'explicitation des choix du décodeur.

5.2.2 Questions de forme

L'utilisation de dictionnaire pose, on l'a vu, plusieurs types de problèmes dans le cadre des modèles neuronaux. L'un d'entre eux concerne la variation morphologique des entrées du dictionnaire, qui peut être source de difficultés (un peu différentes) lorsque le language source ou cible est morphologiquement complexe. Ce problème n'est explicitement traité dans aucune des études que nous avons étudiées ici, qui se limitent à trois alternative assez simplistes : (a) la recopie verbatim de l'entrée source dans la cible ; (b) l'utilisation du décodeur standard pour choisir les entrées à insérer, lorsqu'elles figurent dans son ensemble d'unités de sorties ; (c) la constitution de lexiques associant des formes fléchies et non formes canoniques à partir de corpus, ce qui assure qu'au moins l'appariemment source / cible est cohérent.

La question de la variation morphologique en traduction automatique est souvent traitée en recourant à des modèles d'unités sous-lexicales (Sennrich *et al.*, 2016b; Kudo et Richardson, 2018), voire à des modèles de caractères (Costa-jussà et Fonollosa, 2016; Cherry *et al.*, 2018; Kreutzer et Sokolov, 2018). Faute d'analyse plus poussée des erreurs de ce type, il faut penser que l'utilisation d'unités sous-lexicales (pour autant qu'elles soient cohérentes avec les entrées du dictionnaire) permet de minimiser ce problème, à défaut de le résoudre complètement. Le fait que l'anglais (en source ou en cible) soit

impliqué dans toutes les expérimentations réalisées à ce jour est l'une des raisons pour lesquelles ces problèmes de variation morphologique ne sont pas plus étudiées.

5.3 Méthodes alternatives de décodage

Plusieurs des propositions de ce rapport proposent d'intervenir lors du décodage, ou en aval, pour suggérer (avec plus ou moins de vigueur) le remplacement de l'hypothèse du décodeur par une alternative qui permet de satisfaire la contrainte. Il s'agit donc d'adapter un décodeur standard (gauche-droit) à la prise en compte des contraintes, plutôt que construire le décodeur autour des contraintes qui doivent être imposées. La seule étude qui s'écarte un peu de ce principe est l'étude de Chatterjee *et al.* (2017) qui s'intéresse à l'intégration de contraintes durant la post-édition.

La littérature récente a pourtant montré qu'il était possible de remettre en cause ce processus de décodage récurrent et d'y substituer des algorithmes implantant d'autres stratégies de génération (conditionnelle ou non) d'une phrase cible :

— en générant en parallèle, de manière itérative, les unité cibles (Gu *et al.*, 2018; Lee *et al.*, 2018) à la manière d'une recherche locale ;
— à partir d'indications éparses, que l'on complète en décodant en ordre libre (Ghazvininejad *et al.*, 2019), en s'appuyant sur la capacité des décodeurs stochastiques à fournir des hypothèses à partir de contextes partiellement occultés (ou manquants) ;
— en intégrant les contraintes dans des stratégies de post-édition automatique (Gu *et al.*, 2019), qui ont le mérite de pouvoir disposer d'une vision complète de la phrase traduite, permettant ainsi de simplifier la prise en compte des contraintes.

Ces idées sont explorées dans un travail récent Susanto *et al.* (2020) qui propose de contraindre les opérations d'un décodeur fondé sur la recherche locale et implémentant la technique de (Gu *et al.*, 2019). Le décodage est initialisé avec les contraintes souhaitées, et est modifié itérativement par insertion, suppression et délétions, qui ne peuvent toutefois altérer les contraintes insérées par l'utilisateur. Les auteurs parviennent de cette manière à satisfaire toutes les contraintes de décodage, tout en améliorant le score BLEU.

5.3.1 Évaluation

Une autre observation générale concerne l'évaluation des méthodes de contrôle de la traduction. Dans la majorité des études, la métrique de référence reste le score BLEU, et l'analyse consiste simplement à évaluer les variations du score BLEU selon que l'on intègre ou non des contraintes. Comme on l'a discuté, l'utilisation de contraintes peut simultanément conduire à des améliorations et à des dégradations locales de la performance du système, qu'il importe d'analyser séparément. L'étude méthodologique la plus convaincante est Post *et al.* (2019), qui compare plusieurs méthodes non seulement du point de vue des métriques automatiques, mais également du point de vue de leur capacité à appliquer effectivement les contraintes souhaitées, et à produire des traductions (plus) correctes pour les unités qui font l'objet de contraintes. Cette étude distingue également les performances en fonction des différents types de contrôle effectués : recopie de mots invariables ; traduction d'entités nommées etc et constitue en ce sens un modèle à suivre

pour l'analyse des résultats. Dans cette étude, comme pour d'autres (Macketanz *et al.*, 2018), l'utilisation de jeux de test standard minimise le poids des erreurs portant sur la terminologie ou les entités nommées et rend difficile la mesure des effets d'un contrôle terminologique. Comme le notent Scansani *et al.* (2019), les ressources parallèles annotées avec les termes sont rares et n'existent que pour la paire italien-anglais. En s'appuyant sur ces ressources, il est possible de mettre en œuvre des mesures ciblant spécifiquement les questions terminologiques, comme le *Translation Hit Rate* (THR) utilisé par Farajian *et al.* (2018) et qui dénombre le nombre de termes correctement traduits. Le développement de campagnes d'évaluation portant sur la traduction dans des domaines techniques (Bawden *et al.*, 2019), pour lesquels les problèmes terminologiques abondent, pourrait permettre de développer de nouvelles ressources.

Une autre limitation de certaines études listées dans ce document est qu'elles utilisent souvent des contraintes artificielles qui sont « optimales », au sens où elles sont apprises sur les données de test, plutôt que des contraintes réelles émanant de terminologies ou de dictionaires existants. C'est le cas par exemple de Hasler *et al.* (2018), dont le protocole expérimental est repris par Post et Vilar (2018); Song *et al.* (2020) et permet surtout de mesurer si le mécanisme d'application de contraintes est effectif dans un cadre idéal. À l'inverse, Dinu *et al.* (2019) prennent soin d'utiliser des ressources conséquentes, et permettent d'aboutir à des conclusions plus solides sur l'impact de la méthode.

En allant plus loin dans l'analyse, on note que toutes les études se basent sur des métriques automatiques, qui peinent à prendre en compte l'impact des erreurs de traduction sur l'utilisabilité réelle des documents produits automatiquement. Il est pourtant reconnu que les erreurs portant sur les entités nommées ou les termes peuvent avoir des conséquences particulièrement dommageables, ce qui plaide pour le développement de protocoles expérimentaux impliquant un regard humain pour analyser l'effet réel de la prise en compte de contraintes.

5.4 Conclusion

Dans ce rapport, nous avons dressé un large panorama des méthodes proposées dans la littérature dans le but d'intégrer des contraintes de nature lexicale dans un système de traduction neuronale, et, ce faisant, d'en rendre le comportement plus prédictible. Nous avons dans un premier temps présenté les différents contextes dans lesquels un tel contrôle était souhaitable, ainsi que les contraintes associées. Nous avons ensuite présenté un ensemble de méthodes, de plus en plus précises dans leur contrôle, pour intégrer ces contraintes, en montrant que pour une large part d'entre elles, elles s'inspirent de techniques utilisées dans le contexte de la traduction statistique. Nous avons enfin pointé quelques limitations de ces études, à la fois du point de vue des méthodes qu'elles mettent en œuvre, et des évaluations de leurs performances. Un compromis qui reste difficile à réaliser est d'une part le besoin de « forcer la main au décodeur », afin de s'assurer de la bonne prise en compte des contraintes lorsqu'elles doivent s'appliquer, et d'autre part la nécessité d'interférer le moins possible avec les choix du décodeur, qui a été optimisé pour engendrer des phrases cibles correctes, respectant du mieux possible les contraintes syntaxiques et morphologiques de la langue vers laquelle on traduit.

Bibliographie

Sweta Agrawal et Marine Carpuat : Controlling text complexity in neural machine translation. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1549–1564, Hong Kong, China, novembre 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-1166. [cité page 12]

Tamer Alkhouli, Gabriel Bretschner et Hermann Ney : On the alignment problem in multi-head attention-based neural machine translation. *In Proceedings of the Third Conference on Machine Translation : Research Papers*, pages 177–185, Brussels, Belgium, octobre 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6318. [cité page 17], [cité page 32]

Tamer Alkhouli et Hermann Ney : Biasing attention-based recurrent neural networks using external alignment information. *In Proceedings of the Second Conference on Machine Translation*, pages 108–117, Copenhagen, Denmark, septembre 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W17-4711. [cité page 32]

Alexandre Allauzen et François Yvon : Méthodes statistiques pour la traduction automatique. *In* Eric Gaussier et François Yvon, éditeurs : *Modèles Probabilistes pour l'accès à l'information*, chapitre 7, pages 271–356. Hermès, Paris, 2011. [cité page 5]

Peter Anderson, Basura Fernando, Mark Johnson et Stephen Gould : Guided open vocabulary image captioning with constrained beam search. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1098. [cité page 28]

Philip Arthur, Graham Neubig et Satoshi Nakamura : Incorporating discrete translation lexicons into neural machine translation. *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas, novembre 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D16-1162. [cité page 13], [cité page 27]

Amittai Axelrod, Xiaodong He et Jianfeng Gao : Domain adaptation via pseudo in-domain data selection. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Edinburgh, United Kingdom, 2011. ISBN 978-1-937284-11-4. URL https://www.aclweb.org/anthology/D11-1033.pdf. [cité page 11]

Dzmitry Bahdanau, Kyunghyun Cho et Yoshua Bengio : Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL http://arxiv.org/abs/1409.0473. [cité page 6], [cité page 28]

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor et Maika Vicente Navarro : Findings of the WMT 2019 biomedical translation shared task : Evaluation for MEDLINE abstracts and biomedical terminologies. *In Proceedings of the Fourth Conference on Machine Translation (Volume 3 : Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy, août 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-5403. [cité page 37]

Nicola Bertoldi, Mauro Cettolo et Marcello Federico : Cache-based online adaptation for machine translation enhanced computer assisted translation. *In Proceedings of the Machine Translation Summit*, MT Summit XIV, pages 35–42, Nice, France, 2013. URL http://www.mtsummit2013.info/files/proceedings/main/mt-summit-2013-bertoldi-et-al.pdf. [cité page 16]

Dhouha Bouamor, Nasredine Semmar et Pierre Zweigenbaum : Identifying bilingual multi-word expressions for statistical machine translation. *In Proceedings of the Language Ressource and Evaluation Conference*, LREC'12, pages 674–679, Istambul, Turkey, 2012. URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/886_Paper.pdf. [cité page 16]

Denny Britz, Quoc Le et Reid Pryzant : Effective domain mixing for neural machine translation. *In Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark, 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/W17-4712. [cité page 35]

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer et Paul S. Roossin : A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. URL https://www.aclweb.org/anthology/J90-2002. [cité page 5], [cité page 32]

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra et Robert L. Mercer : The mathematics of statistical machine translation : Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993. URL https://www.aclweb.org/anthology/J93-2003. [cité page 5], [cité page 7]

Franck Burlot : Lingua custodia at WMT'19 : Attempts to control terminology. *In Proceedings of the Fourth Conference on Machine Translation (Volume 2 : Shared Task Papers, Day 1)*, pages 147–154, Florence, Italy, août 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-5310. [cité page 17]

Franck Burlot et François Yvon : Using monolingual data in neural machine translation : a systematic study. *In Proceedings of the Third Conference on Machine Translation : Research Papers*, pages 144–155, Brussels, Belgium, octobre 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6315. [cité page 11]

Marine Carpuat et Mona Diab : Task-based evaluation of multiword expressions : a pilot study in statistical machine translation. *In Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 242–245, Los Angeles, California, juin 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N10-1029. [cité page 15], [cité page 16]

Marine Carpuat et Michel Simard : The trouble with SMT consistency. *In Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 442–449, Montréal, Canada, juin 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W12-3156. [cité page 13]

Rich Caruana : Multitask learning. *Machine Learning*, 28(1):41–75, juillet 1997. ISSN 0885-6125. URL https://doi.org/10.1023/A:1007379606734. [cité page 35]

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia et Frédéric Blain : Guiding neural machine translation decoding with external knowledge. *In Proceedings of the second Conference on Machine Translation*, pages 157–168. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/W17-4716. [cité page 17], [cité page 28], [cité page 36]

Wenhu Chen, Evgeny Matusov, Shahram Khadivi et Jan-Thorsten Peter : Guided alignment training for topic-aware neural machine translation. *In* Spence Green et Lane Schwartz, éditeurs : *Proceedings of the Association of Machine Translation in the Americas*, pages 121–134, Austin, Texas, 2016. [cité page 30]

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat et Wolfgang Macherey : Revisiting character-based neural machine translation with capacity and compression. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium, octobre-novembre 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1461. [cité page 35]

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau et Yoshua Bengio : On the properties of neural machine translation : Encoder-decoder approaches. *In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W14-4012. [cité page 6]

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk et Yoshua Bengio : Learning phrase representations using RNN encoder–decoder for statistical machine translation. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, pages 1724–1734, Doha, Qatar, 2014b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D14-1179. [cité page 6]

Chenhui Chu, Raj Dabre et Sadao Kurohashi : An empirical comparison of domain adaptation methods for neural machine translation. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, ACL 2017, pages 385–391, Vancouver, Canada, 2017. URL http://aclweb.org/anthology/P17-2061. [cité page 11], [cité page 12]

Chenhui Chu et Rui Wang : A survey of domain adaptation for neural machine translation. *In Proceedings of the 27th International Conference on Computational Linguistics*, COLING 2018, pages 1304–1319, Santa Fe, New Mexico, USA, 2018. URL http://aclweb.org/anthology/C18-1111. [cité page 11]

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer et Gholamreza Haffari : Incorporating structural alignment biases into an attentional neural translation model. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 876–885, San Diego, California, juin 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N16-1102. [cité page 7], [cité page 9], [cité page 30]

Alexis Conneau et Guillaume Lample : Cross-lingual language model pretraining. *In* H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox et R. Garnett, éditeurs : *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf. [cité page 35]

Marta R. Costa-jussà et José A. R. Fonollosa : Character-based Neural Machine Translation. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 357–361, Berlin, Germany, août 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P16-2058. [cité page 13], [cité page 35]

Josep Maria Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou et Peter Zoldan : Systran's pure neural machine translation systems. *CoRR*, abs/1610.05540. URL http://arxiv.org/pdf/1610.05540. [cité page 18]

Anna Currey, Antonio Valerio Miceli Barone et Kenneth Heafield : Copied monolingual data improves low-resource neural machine translation. *In Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark, septembre 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W17-4715. [cité page 17]

Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov et Shahram Khadivi : Neural machine translation leveraging phrase-based models in a hybrid search. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1411–1420, Copenhagen, Denmark, septembre 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1148. [cité page 24], [cité page 27], [cité page 29]

Shuoyang Ding, Hainan Xu et Philipp Koehn : Saliency-driven word alignment interpretation for neural machine translation. *In Proceedings of the Fourth Conference on Machine Translation (Volume 1 : Research Papers)*, pages 1–12, Florence, Italy, août 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-5201. [cité page 8]

Georgiana Dinu, Prashant Mathur, Marcello Federico et Yaser Al-Onaizan : Training neural machine translation to apply terminology constraints. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy, juillet 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1294. [cité page 21], [cité page 22], [cité page 37]

Kevin Duh, Graham Neubig, Katsuhito Sudoh et Hajime Tsukada : Adaptation data selection using neural language models : Experiments in machine translation. *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 678–683, Sofia, Bulgaria, 2013. Association for Computational Linguistics. URL http://aclweb.org/anthology/P13-2119. [cité page 11]

Nadir Durrani, Helmut Schmid, Alexander Fraser, Philipp Koehn et Hinrich Schütze : The operation sequence Model—Combining n-gram-based and phrase-based statistical machine translation. *Computational Linguistics*, 41(2):157–186, juin 2015. URL https://www.aclweb.org/anthology/J15-2001. [cité page 35]

Sergey Edunov, Alexei Baevski et Michael Auli : Pre-trained language model representations for language generation. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4052–4059, Minneapolis, Minnesota, juin 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1409. [cité page 35]

M. Amin Farajian, Nicola Bertoldi, Matteo Negri, Marco Turchi et Marcello Federico : Evaluation of terminology translation in instance-based neural mt adaptation. *In* Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Miquel Esplà-Gomis, Maja Popović, Celia Rico, André Martins, Joachim Van den Bogaert et Mikel L. Forcada, éditeurs : *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, EAMT, pages 149–

158, Alicant, Spain, 2018. URL https://rua.ua.es/dspace/bitstream/10045/76037/1/EAMT2018-Proceedings_17.pdf. [cité page 37]

Yang Feng, Shiyue Zhang, Andi Zhang, Dong Wang et Andrew Abel : Memory-augmented neural machine translation. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1390–1399, Copenhagen, Denmark, septembre 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1146. [cité page 17], [cité page 24]

Orhan Firat, Kyunghyun Cho et Yoshua Bengio : Multi-way, multilingual neural machine translation with a shared attention mechanism. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 866–875. Association for Computational Linguistics, 2016. URL http://www.aclweb.org/anthology/N16-1101. [cité page 20]

George Foster et Roland Kuhn : Mixture-model adaptation for SMT. *In Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic, 2007. URL http://www.aclweb.org/anthology/W/W07/W07-0717. [cité page 11]

Markus Freitag et Yaser Al-Onaizan : Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897, 2016. URL http://arxiv.org/abs/1612.06897. [cité page 11]

Kuzman Ganchev, João Graça, Jennifer Gillenwater et Ben Taskar : Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 11:2001–2049, août 2010. ISSN 1532-4435. URL http://jmlr.csail.mit.edu/papers/volume11/ganchev10a/ganchev10a.pdf. [cité page 25]

Juri Ganitkevitch, Benjamin Van Durme et Chris Callison-Burch : PPDB : The paraphrase database. *In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 758–764, Atlanta, Georgia, juin 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1092. [cité page 26]

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy et Matthias Paulik : Jointly learning to align and translate with transformer models. *In Proceedings of the 2019 Conference on EMNLP and the 9th IJCNLP Processing*, Hong Kong, China, novembre 2019. URL https://www.aclweb.org/anthology/D19-1453. [cité page 30]

Hamidreza Ghader et Christof Monz : What does attention in neural machine translation pay attention to ? *In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, pages 30–39, Taipei, Taiwan, novembre 2017. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-1004. [cité page 7], [cité page 28]

Marjan Ghazvininejad, Omer Levy, Yinhan Liu et Luke Zettlemoyer : Mask-predict : Parallel decoding of conditional masked language models. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China, novembre 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-1633. [cité page 36]

Roman Grundkiewicz et Kenneth Heafield : Neural machine translation techniques for named entity transliteration. *In Proceedings of the Seventh Named Entities Workshop*, pages 89–94, Melbourne, Australia, juillet 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-2413. [cité page 14]

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li et Richard Socher : Non-autoregressive neural machine translation. *In Proceedings of the International Conference on Representation Learning*, ICLR'18, 2018. URL http://arxiv.org/abs/1711.02281. [cité page 36]

Jiatao Gu, Zhengdong Lu, Hang Li et Victor O.K. Li : Incorporating copying mechanism in sequence-to-sequence learning. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1631–1640, Berlin, Germany, août 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P16-1154. [cité page 22], [cité page 26]

Jiatao Gu, Changhan Wang et Junbo Zhao : Levenshtein transformer. *In* H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox et R. Garnett, éditeurs : *Advances in Neural Information Processing Systems 32*, pages 11181–11191. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9297-levenshtein-transformer.pdf. [cité page 36]

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou et Yoshua Bengio : Pointing the unknown words. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 140–149, Berlin, Germany, août 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P16-1014. [cité page 26]

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho et Yoshua Bengio : On integrating a language model into neural machine translation. *Comput. Speech Lang.*, 45(C):137–148, septembre 2017. ISSN 0885-2308. URL https://doi.org/10.1016/j.csl.2017.01.014. [cité page 11], [cité page 12]

Dong Han, Junhui Li, Yachao Li, Min Zhang et Guodong Zhou : Explicitly modeling word translations in neural machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(1), juillet 2019. ISSN 2375-4699. URL https://doi.org/10.1145/3342353. [cité page 20]

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias et Bill Byrne : Neural machine translation decoding with terminology constraints. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-2081. [cité page 13], [cité page 28], [cité page 29], [cité page 31], [cité page 37]

Chris Hokamp et Qun Liu : Lexically constrained decoding for sequence generation using grid beam search. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1535–1546. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/P17-1141. [cité page 17], [cité page 22]

Ann Irvine, John Morgan, Marine Carpuat, Hal Daumé et Dragos Munteanu : Measuring machine translation errors in new domains. *Transactions of the Association for Computational Linguistics*, 1:429–440, 2013. URL https://doi.org/10.1162/tacl_a_00239. [cité page 11]

Nal Kalchbrenner et Phil Blunsom : Recurrent continuous translation models. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP'13, pages 1700–1709, Seattle, Washington, USA, 2013. URL http://aclweb.org/anthology/D13-1176. [cité page 6]

Yoon Kim, Carl Denton, Luong Hoang et Alexander M. Rush : Structured attention networks. *In Proceedings of the 5th International Conference on Learning Representations*, (ICLR), Toulon, France, 2017. URL https://openreview.net/forum?id=HkE0Nvqlg. [cité page 32]

Rebecca Knowles et Philipp Koehn : Context and copying in neural machine translation. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3034–3041, Brussels, Belgium, octobre-novembre 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1339. [cité page 27]

Catherine Kobus, Josep Crego et Jean Senellart : Domain control for neural machine translation. *In Proceedings of the International Conference Recent Advances in Natural Language Processing*, RANLP 2017, pages 372–378, Varna, Bulgaria, 2017. URL https://doi.org/10.26615/978-954-452-049-6_049. [cité page 20]

Philip Koehn : *Neural Machine Translation*. Cambridge University Press, 2020. [cité page 5]

Philipp Koehn : *Statistical Machine Translation*. Cambridge University Press, 2010. [cité page 5], [cité page 29]

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard

Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin et Evan Herbst : Moses : Open source toolkit for statistical machine translation. *In Proc. Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, pages 177–180, Prague, Czech Republic, 2007. [cité page 5], [cité page 15]

Philipp Koehn et Rebecca Knowles : Six challenges for neural machine translation. *In Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/W17-3204. [cité page 7], [cité page 28]

Philipp Koehn, Franz Josef Och et Daniel Marcu : Statistical phrase-based translation. *In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistic*, pages 127–133, Edmondton, Canada, 2003. [cité page 5]

Julia Kreutzer et Artem Sokolov : Learning to segment inputs for NMT favors character-level processing. *In Proceedings of the International Workshop on Spoken Language Translation*, IWSLT'18, 2018. URL https://arxiv.org/abs/1810.01480. [cité page 35]

Taku Kudo et John Richardson : SentencePiece : A simple and language independent subword tokenizer and detokenizer for neural text processing. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, pages 66–71, Brussels, Belgium, novembre 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-2012. [cité page 13], [cité page 35]

Roland Kuhn et Renato DeMori : A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 (6):570–583, 1990. [cité page 16]

Jason Lee, Elman Mansimov et Kyunghyun Cho : Deterministic non-autoregressive neural sequence modeling by iterative refinement. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium, octobre-novembre 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1149. [cité page 36]

Huayang Li, Guoping Huang et Lemao Liu : Neural machine translation with noisy lexical constraints, 2019a. [cité page 24]

Xiaoqing Li, Jinghui Yan, Jiajun Zhang et Chengqing Zong : Neural name translation improves neural machine translation. *In China Workshop on Machine Translation*, pages 93–100. Springer, 2018. [cité page 19]

Xintong Li, Guanlin Li, Lemao Liu, Max Meng et Shuming Shi : On the word alignment from neural machine translation. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy,

juillet 2019b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1124. [cité page 8]

Lemao Liu, Masao Utiyama, Andrew Finch et Eiichiro Sumita : Neural machine translation with supervised attention. *In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, pages 3093–3102, Osaka, Japan, décembre 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1291. [cité page 30]

Minh-Thang Luong et Christopher D. Manning : Stanford neural machine translation systems for spoken language domain. *In Proceedings of the International Workshop on Spoken Language Translation*, IWSLT, Da Nang, Vietnam, 2015. [cité page 11]

Minh-Thang Luong et Christopher D. Manning : Achieving open vocabulary neural machine translation with hybrid word-character models. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1054–1063, Berlin, Germany, août 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P16-1100. [cité page 13]

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals et Wojciech Zaremba : Addressing the rare word problem in neural machine translation. *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, pages 11–19, Beijing, China, juillet 2015. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P15-1002. [cité page 18]

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt et Hans Uszkoreit : Fine-grained evaluation of German-English machine translation based on a test suite. *In Proceedings of the Third Conference on Machine Translation : Shared Task Papers*, pages 578–587, Belgium, Brussels, octobre 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6436. [cité page 37]

Jonathan Mallinson et Mirella Lapata : Controllable sentence simplification : Employing syntactic and lexical constraints, 2019. URL https://arxiv.org/abs/1910.04387. [cité page 20]

Kelly Marchisio, Jialiang Guo, Cheng-I Lai et Philipp Koehn : Controlling the reading level of machine translation output. *In Proceedings of Machine Translation Summit XVII Volume 1 : Research Track*, pages 193–2004, Dublin, Ireland, 2019. European Association for Machine Translation. URL https://www.aclweb.org/anthology/W19-6619. [cité page 12]

Fandong Meng, Deyi Xiong, Wenbin Jiang et Qun Liu : Modeling term translation for document-informed machine translation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 546–556, Doha,

Qatar, octobre 2014. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D14-1060. [cité page 13], [cité page 16]

Haitao Mi, Baskaran Sankaran, Zhiguo Wang et Abe Ittycheriah : Coverage embedding models for neural machine translation. *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas, novembre 2016a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D16-1096. [cité page 9]

Haitao Mi, Zhiguo Wang et Abe Ittycheriah : Supervised attentions for neural machine translation. *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, Texas, novembre 2016b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D16-1249. [cité page 30]

Kenton Murray et David Chiang : Correcting length bias in neural machine translation. *In Proceedings of the Third Conference on Machine Translation : Research Papers*, pages 212–223, Brussels, Belgium, octobre 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6322. [cité page 10]

Toan Nguyen et David Chiang : Improving lexical choice in neural machine translation. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, pages 334–343, New Orleans, Louisiana, juin 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N18-1031. [cité page 27]

Jan Niehues, Eunah Cho, Thanh-Le Ha et Alex Waibel : Pre-translation for neural machine translation. *In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, pages 1828–1836, Osaka, Japan, décembre 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1172. [cité page 21]

Xing Niu, Marianna Martindale et Marine Carpuat : A study of style in machine translation : Controlling the formality of machine translation output. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2814–2819, Copenhagen, Denmark, septembre 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1299. [cité page 12], [cité page 20]

Franz Josef Och et Hermann Ney : Discriminative training and maximum entropy models for statistical machine translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, juillet 2002. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P02-1038. [cité page 5]

Franz Josef Och et Hermann Ney : A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003. URL https://www.aclweb.org/anthology/J03-1002. [cité page 7]

Chan Young Park et Yulia Tsvetkov : Learning to generate word- and phrase-embeddings for efficient phrase-based neural machine translation. *In Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 241–248, Hong Kong, novembre 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-5626. [cité page 29]

Ngoc-Quan Pham, Jan Niehues et Alexander Waibel : Towards one-shot learning for rare-word translation with external experts. *In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 100–109, Melbourne, Australia, juillet 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-2712. [cité page 22]

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger et Peyman Passban : Investigating backtranslation in neural machine translation. *In Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, EAMT, Alicante, Spain, 28–30 May 2018. URL https://arxiv.org/abs/1804.06189. [cité page 11]

Matt Post, Shuoyang Ding, Marianna Martindale et Winston Wu : An exploration of placeholding in neural machine translation. *In Proceedings of Machine Translation Summit XVII Volume 1 : Research Track*, pages 182–192, Dublin, Ireland, 2019. European Association for Machine Translation. URL https://www.aclweb.org/anthology/W19-6618. [cité page 18], [cité page 19], [cité page 20], [cité page 36]

Matt Post et David Vilar : Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N18-1119. [cité page 17], [cité page 23], [cité page 24], [cité page 31], [cité page 37]

Shuo Ren, Zhirui Zhang, Shujie Liu, Ming Zhou et Shuai Ma : Unsupervised neural machine translation with SMT as posterior regularization. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 241–248, 2019. URL https://www.aaai.org/ojs/index.php/AAAI/article/view/3791/3669. [cité page 25]

Matīss Rikters et Ondřej Bojar : Paying attention to multi-word expressions in neural machine translation, 2017. URL http://arxiv.org/pdf/1710.06313. [cité page 18]

Randy Scansani, Luisa Bentivogli, Silvia Bernardini et Adriano Ferraresi : MAGMATic : A multi-domain academic gold standard with manual annotation of

terminology for machine translation evaluation. *In Proceedings of Machine Translation Summit XVII Volume 1 : Research Track*, pages 78–86, Dublin, Ireland, août 2019. European Association for Machine Translation. URL https://www.aclweb.org/anthology/W19-6608. [cité page 12], [cité page 37]

Rico Sennrich et Barry Haddow : Linguistic input features improve neural machine translation. *In Proceedings of the First Conference on Machine Translation : Volume 1, Research Papers*, pages 83–91, Berlin, Germany, août 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W16-2209. [cité page 20]

Rico Sennrich, Barry Haddow et Alexandra Birch : Improving neural machine translation models with monolingual data. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 86–96, Berlin, Germany, August 2016a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1009. [cité page 11]

Rico Sennrich, Barry Haddow et Alexandra Birch : Neural machine translation of rare words with subword units. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1715–1725, Berlin, Germany, août 2016b. URL https://www.aclweb.org/anthology/P16-1162. [cité page 8], [cité page 13], [cité page 35]

Kai Song, Kun Wang, Heng Yu, Yue Zhang, Zhongqiang Huang, Weihua Luo, Xiangyu Duan et Min Zhang : Alignment-enhanced transformer for constraining nmt with pre-specified translations. *In Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, 2020. AAAI. URL https://www.aaai.org/Papers/AAAI/2020GB/AAAI-SongK.6422.pdf. [cité page 17], [cité page 30], [cité page 31], [cité page 32], [cité page 37]

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang et Min Zhang : Code-switching for enhancing NMT with pre-specified translation. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota, juin 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1044. [cité page 21]

Felix Stahlberg et Bill Byrne : On NMT search errors and model errors : Cat got your tongue ? *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China, novembre 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-1331. [cité page 10]

Felix Stahlberg, James Cross et Veselin Stoyanov : Simple fusion : Return of the language model. *In Proceedings of the Third Conference on Machine Translation :*

Research Papers, pages 204–211, Brussels, Belgium, octobre 2018a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6321. [cité page 11]

Felix Stahlberg, Danielle Saunders et Bill Byrne : An operation sequence model for explainable neural machine translation. *In Proceedings of the 2018 EMNLP*, pages 10–21, Brussels, Belgium, novembre 2018b. URL https://www.aclweb.org/anthology/K16-1002. [cité page 13], [cité page 35]

Raymond Hendy Susanto, Shamil Chollampatt et Liling Tan : Lexically constrained neural machine translation with Levenshtein transformer. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online, juillet 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.325. [cité page 36]

Liling Tan, Josef van Genabith et Francis Bond : Passive and pervasive use of bilingual dictionary in statistical machine translation. *In Proceedings of the Fourth Workshop on Hybrid Approaches to Translation (HyTra)*, pages 30–34, Beijing, juillet 2015. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W15-4105. [cité page 18]

Yaohua Tang, Fandong Meng, Zhengdong Lu, Hang Li et Philip L. H. Yu : Neural machine translation with external phrase memory. *ArXiv*, abs/1606.01792, 2016. [cité page 29]

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu et Hang Li : Modeling coverage for neural machine translation. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 76–85, Berlin, Germany, août 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P16-1008. [cité page 9], [cité page 16]

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser et Illia Polosukhin : Attention is all you need. *In* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et R. Garnett, éditeurs : *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017a. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf. [cité page 6], [cité page 28]

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser et Illia Polosukhin : Attention is all you need. *In* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et R. Garnett, éditeurs : *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017b. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf. [cité page 8]

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong et Min Zhang : Neural machine translation advised by statistical machine translation. *In Thirty-First*

AAAI Conference on Artificial Intelligence, pages 3330–336, San Francisco, CA, USA, 2017a. AAAI. URL https://arxiv.org/pdf/1610.05150v1.pdf. [cité page 24]

Xing Wang, Zhaopeng Tu, Deyi Xiong et Min Zhang : Translating phrases in neural machine translation. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1421–1431, Copenhagen, Denmark, septembre 2017b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1149. [cité page 24], [cité page 29]

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey *et al.* : Google's neural machine translation system : Bridging the gap between human and machine translation, 2016. URL https://arxiv.org/abs/1609.08144. [cité page 9], [cité page 10]

Jiacheng Zhang, Yang Liu, Huanbo Luan, Jingfang Xu et Maosong Sun : Prior knowledge integration for neural machine translation using posterior regularization. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1514–1523, Vancouver, Canada, juillet 2017a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P17-1139. [cité page 25]

Jiajun Zhang et Chengqing Zong : Bridging neural machine translation and bilingual dictionaries. *arXiv preprint arXiv :1610.07272*, 2016. URL http://arxiv.org/pdf/1610.07272.pdf. [cité page 18]

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig et Satoshi Nakamura : Improving neural machine translation through phrase-based forced decoding. *In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, pages 152–162, Taipei, Taiwan, novembre 2017b. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-1016. [cité page 29]

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono et Bambang Parmanto : Integrating transformer and paraphrase rules for sentence simplification. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium, 2018a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1355. [cité page 26]

Yang Zhao, Yining Wang, Jiajun Zhang et Chengqing Zong : Phrase table as recommendation memory for neural machine translation. *In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI-18, pages 4609–4615. International Joint Conferences on Artificial Intelligence Organization, 7 2018b. URL https://doi.org/10.24963/ijcai.2018/641. [cité page 24], [cité page 27]

Qingyu Zhou, Nan Yang, Furu Wei et Ming Zhou : Sequential copying networks. *In Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI 2018, 2018. URL

https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16323/16032. [cité page 27]