Université de Montréal

Extraction de phrases parallèles à partir d'un corpus comparable avec des réseaux de neurones récurrents bidirectionnels

par

Francis Grégoire

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique

14 décembre 2017

SOMMAIRE

Les corpus parallèles sont cruciaux pour le bon fonctionnement des applications multilingues du traitement automatique du langage naturel. Comme ils sont des ressources essentielles, le nombre limité de corpus parallèles, que nous trouvons pour un nombre relativement faible de paires de langues sur très peu de domaines spécifiques, est problématique pour le développement des applications de traitement automatique du langage naturel. L'extraction de phrases parallèles est une tâche qui s'attaque directement au problème de manque de données en extrayant des phrases parallèles depuis l'importante quantité d'articles multilingues retrouvés sur le Web.

Dans ce mémoire, nous proposons un système d'extraction de phrases parallèles qui mesure la relation de traduction entre les phrases dans deux langues. Notre système est une approche basée sur des réseaux de neurones récurrents bidirectionnels qui peut apprendre les représentations des phrases dans un espace vectoriel conjoint en maximisant explicitement la similarité entre les phrases parallèles. Contrairement aux approches précédentes, en exploitant ces représentations vectorielles continues des phrases nous enlevons le besoin d'utiliser plusieurs modèles et toute ingénierie de traits spécifiques. Notre approche s'entraîne directement avec des paires de phrases et s'adapte facilement à une grande quantité de données.

Des expériences sur des corpus parallèles bruités montrent que notre approche surpasse un système de référence à l'état de l'art. Pour justifier l'utilité de notre approche, nous ajoutons les paires de phrases extraites des articles de Wikipédia à un corpus parallèle pour entraîner des systèmes de traduction automatique et nous obtenons une amélioration de la performance de traduction. Nos résultats empiriques nous amènent à croire que notre système est un outil prometteur pour créer de nouvelles ressources multilingues alignées.

Mots-clés : extraction de phrases parallèles, traduction automatique, traitement automatique du langage naturel, apprentissage profond, réseaux de neurones récurrents, corpus parallèle, corpus comparable.

SUMMARY

Parallel corpora are a prerequisite for many multilingual natural language processing applications. As they are an invaluable resource, the limited amount of parallel data, which is only available for a relatively small number of language pairs on very few specific domains, is problematic for scaling natural language processing applications. Parallel sentence extraction is a task addressing the data sparsity problem by extracting parallel sentences from the increasing amount of content-related multilingual articles on the World Wide Web.

In this thesis, we propose a parallel sentence extraction system to measure the translational equivalence between sentences in two languages. Our system is a bidirectional recurrent neural network based approach that can learn sentence representations in a shared vector space by explicitly maximizing the similarity between parallel sentences. In contrast to previous approaches, by leveraging these continuous vector representation of sentences we remove the need to rely on multiples models and any specific feature engineering. Our approach can be efficiently trained with raw sentence pairs and is scalable to large amount of data.

Experiments on noisy parallel corpora show that our approach outperforms a state-of-the-art baseline. To justify the utility of our approach, we add the sentence pairs extracted from Wikipedia articles to a parallel corpus to train machine translation systems and show improvement in translation performance. Our experimental results lead us to believe that our system is a promising tool to create new aligned multilingual resources.

Keywords : parallel sentence extraction, machine translation, natural language processing, deep learning, recurrent neural networks, parallel corpora, comparable corpora.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET DES ABRÉVIATIONS

BiRNN	Réseaux de neurones récurrents bidirectionnels, de l'anglais *Bidirectional Recurrent Neural Networks*
GPU	Processeur graphique, de l'anglais *Graphics Processing Unit*
GRU	De l'anglais *Gated Recurrent Unit*
HMM	Modèle de Markov caché, de l'anglais *Hidden Markov Model*
LSTM	De l'anglais *Long Short Term Memory*
NMT	Traduction automatique neuronal, de l'anglais *Neural Machine Translation*
RNN	Réseau de neurones récurrents, de l'anglais *Recurrent Neural Network*
SGD	Algorithme du gradient stochastique, de l'anglais *Stochastic Gradient Descent*
SMT	Traduction automatique statistique, de l'anglais *Statistical Machine Translation*
TALN	Traitement automatique du langage naturel
k	Symbole pour désigner 10^3
M	Symbole pour désigner 10^6

DÉDICACES

À mes parents.

INTRODUCTION

Motivation

Les corpus parallèles sont des ressources fondamentales pour le bon fonctionnement des applications multilingues du domaine du traitement automatique du langage naturel (TALN). Un corpus parallèle pour une paire de langues est une collection de textes dans deux langues avec des phrases alignées qui sont la traduction l'une et de l'autre. À ce jour, nous retrouvons des corpus parallèles de bonne qualité pour un nombre relativement restreint de paires de langues sur très peu de domaines spécifiques. Ces sources de données sont souvent des procédures parlementaires de pays traduites en plusieurs langues. Les corpus parallèles les plus connus et les plus utilisés sont les procédures parlementaires du parlement européen traduites dans 21 langues [1]. Ces données sont alignées par rapport aux textes en anglais et n'ont pas été mises à jour depuis 2012. Chaque paire de langue n'a pas le même nombre de paires de phrases parallèles. Par exemple, le corpus parallèle anglais-français comporte environ 2M de paires de phrases, alors que celui anglais-roumain n'en a que 400k. De plus, seulement deux langues européennes se trouvent parmi les cinq langues les plus parlées dans le monde, qui sont respectivement le chinois, l'espagnol, l'anglais, l'hindi et l'arabe. Toutefois, la qualité des phrases de ces corpus est garantie car les traductions ont été faites par des traducteurs professionnels. Depuis les dernières années, des efforts continus ont été faits pour regrouper des textes parallèles à partir de nouvelles sources de données. La collection OPUS [2], une plateforme rassemblant des textes parallèles de plusieurs sources diverses, en est un bon exemple.

Cette limitation dans la quantité de textes parallèles est problématique pour le développement de nouvelles applications du TALN. Le problème le plus important qui arrive lorsque nous avons accès à une quantité limitée de ressources est que nous avons une grande quantité de mots qui sont exclus du vocabulaire des données utilisées pour faire l'entraînement des modèles. Une autre situation fréquemment rencontrée est celle où les données d'entraînement et d'évaluation viennent de différents domaines. Dans une telle

1. http ://www.statmt.org/europarl/
2. http ://opus.nlpl.eu/

situation, la performance d'une application exploitant ces ressources sera souvent limitée. Il faudrait idéalement utiliser des données d'entraînement supplémentaires afin de couvrir les domaines manquants. Malheureusement, cette condition idéale n'est pas réaliste.

Pour ces raisons, nous trouvons de plus en plus d'intérêt dans le domaine à faire de la collection de données parallèles pour accroître les données actuellement disponibles. Avec la progression exponentielle du nombre d'articles dans plusieurs langues que nous trouvons sur le Web, une solution potentielle pour répondre au problème du manque de données est d'utiliser un système dans le but d'identifier et d'extraire les phrases parallèles qui se trouvent dans ces ressources d'information abondantes. Par conséquent, l'objectif de la tâche d'extraction de phrases parallèles est de créer de nouveaux corpus parallèles à partir d'articles ou de documents traitant d'un même sujet dans des langues différentes. Ces corpus parallèles sont utilisés pour élargir la quantité des données et l'ensemble des domaines couverts dans le but d'améliorer la performance des applications multilingues du TALN, dont principalement les systèmes de traduction automatique.

Un système d'extraction de phrases parallèles est typiquement divisé en deux tâches. En premier lieu, un alignement au niveau des documents est réalisé pour associer les paires de documents pertinents. Ensuite, les paires de phrases parallèles sont détectées dans ces paires de documents. Certains travaux se sont concentrés sur la première tâche en proposant des systèmes conçus pour créer des corpus comparables de haute qualité (Chen and Nie, 2000; Otero and López, 2010; Patry and Langlais, 2011), alors que plusieurs autres font de l'extraction de phrases à partir de ces corpus comparables pour générer de nouveaux corpus parallèles (Munteanu and Marcu, 2005; Adafre and de Rijke, 2006; Abdul-Rauf and Schwenk, 2009; Smith et al., 2010; Uszkoreit et al., 2010). Dans ce mémoire nous nous concentrons uniquement sur l'extraction de phrases parallèles.

Il est naturel de se demander dans quelle mesure l'usage des algorithmes traditionnels d'alignement de phrases (Gale and Church, 1993; Moore, 2002; Lamraoui and Langlais, 2013) n'est pas suffisant pour identifier les paires de phrases parallèles à partir de tels corpus comparables ? Ces systèmes ont été conçus pour aligner les phrases des corpus parallèles de manière monotone, c'est-à-dire en supposant qu'il n'y a pas de réordonnancement des phrases dans l'une des langues. Ainsi, ces algorithmes ne sont pas adaptés à des textes hautement non parallèles.

Par définition, un corpus comparable contient des textes dans plusieurs langues dont les sujets sont alignés, mais qui ne sont pas traduits. S'il s'y trouvent des paires de phrases en

relation de traduction, le niveau d'alignement varie selon leur degré de comparabilité. Idéalement, si nous voulons bien comparer des textes, nous en voulons qui soient étroitement liés et qui décrivent le même sujet. Un corpus comparable fréquemment utilisé dans la recherche et la pratique est l'encyclopédie collaborative en ligne Wikipédia [3]. En tant qu'encyclopédie, il est possible d'extraire des phrases de domaines spécifiques ayant une morphologie et un vocabulaire différents que ceux couverts dans les corpus parallèles populaires. De plus, plusieurs articles sont disponibles dans plusieurs langues et peuvent souvent être connectés avec des liens interlangues. Bien que ces liens nous permettent d'associer des articles dans plusieurs langues qui traitent d'un même sujet et d'éviter le besoin d'utiliser un modèle d'alignement de documents, cela n'empêche pas qu'en faire l'extraction de phrases parallèles reste une tâche difficile. En effet, nous ne savons pas comment l'information est partagée entre les différents articles dans plusieurs langues. Certaines paires d'articles peuvent être des traductions l'une de l'autre, alors que la majorité peut être très peu comparable.

D'un autre côté, les progrès en apprentissage profond au cours des dernières années ont permis de développer des modèles d'apprentissage automatique produisant des résultats qui sont maintenant considérés comme l'état de l'art dans plusieurs domaines, tels que la reconnaissance d'image et le traitement de la parole. Plus récemment, les avancées des modèles à base de réseaux neurones ont commencées à être appliquées aux applications textuelles du TALN. En particulier, les réseaux de neurones récurrents (RNNs) (*recurrent neural networks*, en anglais) ont montré qu'ils peuvent apprendre à convertir des séquences de longueurs variables en représentations vectorielles continues de tailles fixes, permettant d'avoir des résultats très prometteurs. Les RNNs ont été appliqués avec succès à de nombreuses tâches du TALN, comme par exemple la génération de l'écriture manuscrite (Graves, 2013), la génération de texte à partir d'image (Vinyals et al., 2014; Xu et al., 2015), la génération de dialogue (Sordoni et al., 2015; Vinyals and Le, 2015) et les systèmes question réponse (Yu et al., 2014; Hermann et al., 2015). La plupart des efforts pour les applications multilingues ont été consacrés à la traduction automatique neuronale (NMT) (*neural machine translation*, en anglais) (Sutskever et al., 2014; Cho et al., 2014).

Les systèmes d'extraction de phrases parallèles précédents ont démontré empiriquement que l'ajout des paires de phrases parallèles extraites des corpus comparables améliore les performances des systèmes de traduction automatique statistique (SMT) (*statistical machine translation*, en anglais). Cependant, de telles méthodes reposent sur une quantité importante d'ingénierie de traits (*feature engineering*, en anglais) et elles sont difficiles à adapter sur des données qui ont des contextes différents que ceux retrouvés dans les données d'entraînement. Un défi majeur avec ces approches est que plusieurs paires de phrases parallèles peuvent

3. https ://www.wikipedia.org/

avoir une faible similarité lexicale. Dans ce cas, le contexte devient un facteur clé. De plus, les corpus hors domaine amènent des difficultés supplémentaires dû à la couverture limitée du vocabulaire. Dans ce travail, nous proposons une approche à base de réseaux de neurones récurrents bidirectionnels pour faire l'extraction de phrases parallèles à partir de corpus comparables. Nous allons voir qu'une approche comme celle que nous proposons améliore la qualité des paires de phrases extraites.

Contributions

Les systèmes d'extraction de phrases parallèles développés jusqu'à présent nécessitent des métadonnées ou l'utilisation d'une série de modèles qui ne s'adaptent pas nécessairement bien aux différentes structures des textes. Pour répondre à ce problème, nous montrons que nous pouvons construire des nouveaux corpus parallèles depuis un corpus comparable avec un seul modèle basé sur des réseaux de neurones récurrents. Cette approche élimine le besoin d'utiliser des modèles d'alignement de mots en traitant directement des paires de phrases pour estimer la probabilité que deux phrases soient en relation de traduction. Nous montrons empiriquement que la performance de notre approche surpasse celle d'un système considéré comme l'état de l'art pour cette tâche. Nous obtenons des résultats très prometteurs en enlevant le besoin d'appliquer du *feature engineering* et le besoin d'utiliser des ressources externes. En utilisant les paires de phrases que nous avons extraites d'un corpus comparable pour entraîner des systèmes SMT et NMT, nous montrons qu'elles sont une ressource considérable à exploiter. C'est la première fois que les phrases extraites d'un tel système sont évaluées dans un contexte de traduction avec des systèmes NMT. Au meilleur de notre connaissance, l'apprentissage profond n'avait pas encore été appliqué à la création d'un système d'extraction de phrases parallèles complet.

Organisation du mémoire

Le chapitre 1 fait le survol des systèmes d'extraction de phrases parallèles populaires et de quelques approches connexes et pertinentes qui répondent au problème du manque de données parallèles. Ces approches exploitent les données des corpus comparables et des collections de textes non parallèles dans le but de perfectionner le développement et la performance des systèmes de traduction automatique.

Dans le chapitre 2, nous passons en revue les avancées des modèles à base de réseaux de neurones et de l'apprentissage profond appliqué dans le contexte du TALN. Ces modèles sont essentiels pour comprendre l'approche que nous proposons et le reste du mémoire.

Le chapitre 3 combine les concepts des deux chapitres précédents pour introduire le système d'extraction de phrases parallèles que nous proposons.

Une étude approfondie est faite dans le chapitre 4 où nous présentons les résultats de nos expériences. Nous commençons par évaluer la capacité de notre approche à identifier et extraire des paires de phrases d'un corpus parallèle bruité en la comparant à un système de référence basé sur deux travaux clés du domaine. Par la suite, nous évaluons l'utilité et la pertinence des phrases extraites d'un corpus comparable en les ajoutant à un corpus parallèle pour entraîner des systèmes de traduction automatique. Notre montrons que notre approche est supérieure aux systèmes existants selon divers aspects. Ce chapitre étend l'article (Grégoire and Langlais, 2017) que nous avons produit pour la présentation d'une affiche au Symposium IA Montréal [4].

Le chapitre 5 conclut le mémoire en faisant un résumé du travail accompli et en discutant des pistes de travaux futurs.

4. http ://montrealaisymposium.com/

Chapitre 1

LES SYSTÈMES D'EXTRACTION DE PHRASES PARALLÈLES ET AUTRES APPROCHES

Les différents systèmes de traduction automatique partagent les deux mêmes problèmes fondamentaux :

1. ils nécessitent une très grande quantité de phrases parallèles pour avoir une bonne performance.
2. plusieurs domaines doivent être couverts dans les données d'entraînement pour être en mesure de généraliser à de nouvelles sources de données.

Pour la plupart des paires de langues, les corpus parallèles sont en général une ressource rare, voir inexistante. De plus, si nous désirons des modèles fonctionnels et performants pour des domaines précis, il devient encore plus difficile et coûteux d'obtenir des données parallèles de bonne qualité en quantité suffisante. Cette rareté des données parallèles est problématique car elle limite le développement de la recherche et des applications multilingues en TALN. Pour ces raisons, diverses approches ont été développées à travers les dernières années pour tirer profit des collections de textes retrouvées sur le Web afin d'améliorer les modèles existants.

Dans ce chapitre, nous présentons différentes approches qui ont été proposées à cette fin. Nous commençons par faire un survol des travaux clés sur les systèmes d'extraction de phrases parallèles. Ceux-ci exploitent les données du Web et des corpus comparables dans le but d'augmenter la quantité et les domaines couverts par les ensembles d'entraînement des systèmes de traduction automatique. Ensuite, nous présentons des approches récentes qui exploitent ces données de manières différentes, mais avec un objectif très similaire.

1.1. Systèmes d'extraction de phrases parallèles

Dans (Munteanu and Marcu, 2005), les auteurs présentent un système complet pour extraire automatiquement des paires de phrases d'une collection de journaux en anglais, chinois et

arabe. Les auteurs évaluent la qualité des phrases extraites en montrant qu'elles améliorent la précision d'un système de traduction statistique. Cette approche est considérée comme l'état de l'art et sa structure est encore celle utilisée dans les nouvelles approches. Un aperçu de cette structure, adapté de (Munteanu and Marcu, 2005), est présenté à la Figure 1.1. Le système est divisé en deux parties. En premier, un système de recherche d'information est utilisé pour aligner et sélectionner les paires de documents similaires en utilisant les dates de publication. Ensuite, les phrases parallèles sont détectées dans les paires de documents identifiés. Pour y arriver, des modèles d'alignement IBM 1 (Brown et al., 1993) sont entraînés sur un corpus parallèle pour induire des lexiques bilingues et des tables d'alignement qui serviront à sélectionner les paires de phrases. À partir de chacune de ces paires de documents, toutes les paires de phrases du produit cartésien des deux documents sont acheminées à travers un processus de filtrage pour réduire le nombre de paires de phrases à évaluer. Le filtrage se fait en deux étapes. En comparant la longueur entre les deux phrases, chaque paire avec un ratio de longueur supérieur à deux est retirée. En utilisant les lexiques bilingues induits, un deuxième filtre s'assure ensuite qu'au moins la moitié des mots de chaque phrase ont une traduction dans l'autre phrase. Les paires de phrases candidates sont envoyées à un classificateur d'entropie maximale qui détermine si deux phrases sont en relation de traduction. En utilisant les paires de phrases extraites comme données supplémentaires à un corpus parallèle standard pour faire l'apprentissage des systèmes SMT, les auteurs montrent que cela améliore la performance de traduction.

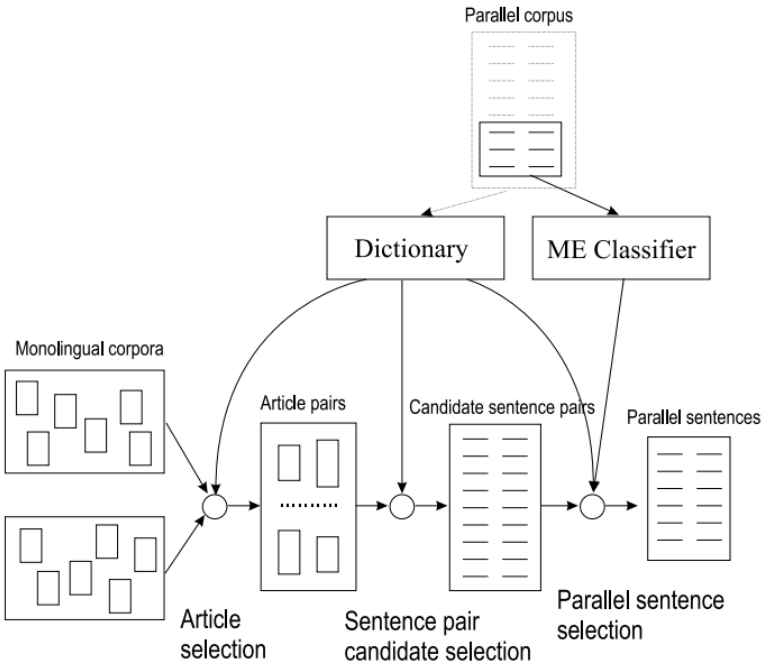


FIGURE 1.1. Système d'extraction de phrases parallèles (Munteanu & Marcu (2005)).

(Adafre and de Rijke, 2006) est le premier travail à chercher si les articles trouvés sur Wikipédia sont une source susceptible de générer des corpus parallèles. Pour y arriver, les deux auteurs utilisent deux approches sur des paires d'articles en anglais et néerlandais. Dans la première, ils utilisent simplement un engin de traduction en ligne pour traduire vers l'anglais un article en néerlandais. La nouvelle paire d'articles est ensuite segmentée en phrases et la similarité de chaque paire de phrases du produit cartésien est mesurée par l'indice de Jaccard sur l'intersection des mots

$$Jaccard(\mathbf{s}_i^s, \mathbf{s}_j^t) = \frac{|\mathbf{s}_i^s \cap \mathbf{s}_j^t|}{|\mathbf{s}_i^s \cup \mathbf{s}_j^t|}, \tag{1.1.1}$$

où $\mathbf{s}_i^s$ est une phrase de l'article en anglais et $\mathbf{s}_j^t$ est une phrase de l'article en néelandais traduit vers l'anglais. La deuxième approche dépend d'un lexique bilingue qui a été induit à partir de la traduction des titres dans la structure des liens interlangues d'un article. Malgré la simplicité de leurs deux heuristiques, ils trouvent que Wikipédia est une excellente source de données pour générer des phrases parallèles.

Au lieu d'employer un classificateur comme dans (Munteanu and Marcu, 2005), (Abdul-Rauf and Schwenk, 2009) suggère d'utiliser un système SMT construit à partir d'un petit corpus parallèle pour traduire des articles en français des corpus comparables LDC Gigaword anglais-français [1]. Une fois un article traduit, les phrases traduites et la date sont utilisées pour identifier des phrases candidates du côté des articles en anglais en utilisant un système de recherche d'information. Les scores du taux d'erreur des mots (*word error rate*, en anglais) et du taux d'erreur de traduction (*translation error rate*, en anglais) sont utilisés entre la traduction d'une phrase source et une phrase cible candidate extraite pour déterminer si elles sont parallèles ou non.

(Smith et al., 2010) étend l'approche de (Munteanu and Marcu, 2005) et de (Adafre and de Rijke, 2006) en exploitant la structure et les métadonnées des paires d'articles Wikipédia comme source d'information additionnelle pour extraire des paires de phrases parallèles. Motivé par l'observation que les paires de phrases parallèles sont souvent trouvées à proximité dans les corpus comparables, ils introduisent des nouvelles *features* qui prennent en compte la position des phrases actuelles et précédemment alignées. Ils utilisent leur ensemble de *features* augmenté dans des champs aléatoires conditionnels (*conditional random fields*, en anglais) et obtiennent des résultats à l'état de l'art sur un petit ensemble de 20 paires d'articles Wikipédia annotés manuellement pour les paires de langues anglais-bulgare, anglais-espagnol et anglais-allemand.

1. Les corpus comparables LDC Gigaword sont des larges collections de textes d'agences d'information journalistique dans plusieurs langues : https ://www.ldc.upenn.edu/

(Barrón-Cedeño et al., 2015) propose une approche sous forme de graphe formé sur la structure des métadonnées de Wikipédia pour faire l'extraction automatique de textes comparables en plusieurs langues sur des sujets spécifiques à partir de Wikipédia. Bien que l'objectif de leur modèle est l'alignement de documents spécifiques à un domaine pour créer des corpus comparables de haute qualité, les auteurs proposent d'extraire des paires de phrases entre deux articles alignés de manière non supervisée. Étant donné une paire d'articles, la similarité entre toutes les paires de phrases est estimée en utilisant différentes mesures de similarité populaires dans la recherche d'information translingue. Malgré qu'avec cette méthode d'extraction ils obtiennent un corpus parallèle bruité avec des scores de précision de rappel relativement faibles, ils observent que les paires de phrases extraites améliorent de manière significative la qualité de la traduction des systèmes SMT évalués sur les domaines spécifiques des phrases extraites.

Le seul système d'extraction de phrases parallèles qui utilise des réseaux de neurones que nous avons trouvé est celui de (Chu et al., 2016). Dans ce travail, les auteurs entraînent un système de traduction neuronal (Bahdanau et al., 2014) sur un corpus parallèle chinois-japonais et ajoutent les représentations vectorielles des phrases de l'encodeur comme *features* additionnels au système de (Munteanu and Marcu, 2005). Au lieu de se servir d'un classificateur d'entropie maximale, une machine à vecteurs de support (*support vector machine*, en anglais) est utilisée comme classificateur. Cette approche est totalement différente de la nôtre, où nous faisons abstraction de tous les modèles de langue et d'alignement externes au classificateur en utilisant un unique modèle bout-à-bout (*end-to-end*, en anglais) à base de réseaux de neurones profonds pour estimer la distribution de probabilité conditonnelle que des paires de phrases soient en relation de traduction.

1.2. Autres approches

Bien que les approches suivantes ne sont pas des systèmes d'extraction de phrases parallèles, nous jugeons qu'elles sont fortement liées en répondant aussi au manque de données parallèles pour faire l'apprentissage d'un système de traduction automatique robuste.

Dans leur travail, (Irvine and Callison-Burch, 2016) développent une approche qui repose uniquement sur des paires de textes non parallèles pour entraîner des systèmes SMT. À la base de cette approche se trouve un modèle d'induction de lexique bilingue basé sur un modèle discriminatif qui leur permet de produire des traductions plus précises que celles des méthodes d'induction précédentes. Au lieu de créer un corpus parallèle comme avec l'extraction de phrases parallèles, la tâche d'induction de lexique bilingue permet de générer de nouveaux lexiques bilingues à partir des corpus comparables et d'un petit lexique bilingue. Les auteurs définissent l'induction de lexique bilingue en un problème de

classification binaire où une paire de mots peut être une traduction ou non. Une variété de signaux d'équivalence de traduction sont utilisés comme *features* pour leur classificateur, tels que des scores de similarité contextuelle, temporelle, orthographique, thématique et fréquentielle. Ce système crée des nouveaux lexiques bilingues qui peuvent servir à générer de nouvelles traductions à partir du décodeur d'un système SMT ou à améliorer les systèmes de traduction automatique lorsque la quantité de données parallèles est limitée.

Toujours en réponse au manque des données parallèles, (Xia et al., 2016) ouvre la porte vers un nouveau paradigme intéressant où un mécanisme à double apprentissage permet aux systèmes de traduction automatique d'apprendre directement sur des textes non parallèles grâce à des méthodes d'apprentissage par renforcement (*reinforcement learning*, en anglais). Le double apprentissage peut être interprété comme un jeu de communication entre deux agents unilingues qui s'échangent des messages à travers un canal bruité dans chaque direction. Ces deux canaux de communication agissent en tant que modèles de traduction. À chaque retour d'information, les agents voient si les canaux de communication fonctionnent bien et peuvent les améliorer en conséquence. Il est démontré que cette approche améliore la performance d'un système NMT, en plus de réduire significativement le besoin d'une grande quantité de données parallèles en entraînement.

La récente recherche de (Lample et al., 2017) étend l'approche de (Irvine and Callison-Burch, 2016) en tirant profit des techniques modernes en apprentissage profond. Les auteurs explorent s'il est possible d'entraîner un système NMT de manière non supervisée où les ressources sont uniquement deux textes non parallèles. L'idée principale est de construire un espace vectoriel conjoint pour les deux langues et d'apprendre à reconstruire des phrases traduites dans les deux sens à partir des représentations vectorielles latentes. Ces représentations vectorielles pour les langues source et cible sont reconstruites par un autoencodeur de débruitage (*denoising autoencoder*, en anglais) (Vincent et al., 2008) et elles sont contraintes à partager la même distribution par apprentissage accusatoire (*adversarial training*, en anglais) où le modèle doit tromper un discriminateur qui est simultanément entraîné pour identifier la langue d'une représentation vectorielle latente.

Chapitre 2

LES MODÈLES À BASE DE RÉSEAUX DE NEURONES

Dans ce chapitre nous allons passer en revue les avancées des modèles à base de réseaux de neurones appliqués au TALN. Nous prenons le temps de les détailler car ces méthodes sont fondamentales au développement et à la compréhension de l'approche que nous proposons au chapitre 3.

2.1. Modèles de langue neuronaux

Une phrase $\mathbf{s}$ peut être représentée comme une séquence de N symboles discrets (ou unités lexicales) tel que $\mathbf{s} = (w_1, \ldots, w_N)$, où w_t est un mot ou un symbole de ponctuation. Chacun de ces symboles est aussi représenté par un nombre entier équivalent à sa position dans le vocabulaire V contenant les symboles d'un langage. L'objectif d'un modèle de langue est d'estimer la probabilité conjointe d'une séquence $p(\mathbf{s}) = p(w_1, \ldots, w_N)$, que nous pouvons décomposer en un produit de probabilités conditionnelles, telle que :

$$p(\mathbf{s}) = \prod_{t=1}^{N} p(w_t | w_1, \ldots, w_{t-1}). \tag{2.1.1}$$

De tels modèles de langue sont utilisés dans de nombreuses applications du TALN, telles que la traduction automatique, la reconnaissance de la parole et la recherche d'information. En réalité, cette distribution n'est pas définie et il n'est pas trivial de l'estimer. Une manière de réduire considérablement la difficulté de ce problème de modélisation consiste à assumer que les mots les plus près dans une séquence sont statistiquement plus dépendants que les mots plus éloignés. C'est en appliquant la propriété de Markov d'ordre $n-1$ qu'un modèle de langue statistique de type n-gramme (*n-gram*, en anglais) nous permet d'approximer la distribution conditionnelle d'un symbole étant donné les $n-1$ symboles précédents, de sorte que :

$$p(w_t | w_1, \ldots, w_{t-1}) \approx p(w_t | w_{t-n+1}, \ldots, w_{t-1}). \tag{2.1.2}$$

Pour entraîner un modèle de langue, nous sommes obligés d'utiliser un corpus où la grande majorité des phrases plausibles dans une langue n'apparaissent pas. Autrement dit, nous devons estimer une distribution sur un ensemble infiniment petit par rapport à l'espace de toutes les phrases d'un langage. Cela fait en sorte que les modèles de langue statistiques (Chen and Goodman, 1999) ne généralisent pas nécessairement de manière efficace. D'un autre côté, plus un modèle de langue statistique est entraîné sur un grand corpus, plus la taille du vocabulaire augmente. En revanche, le nombre de séquences de symboles possibles augmente exponentiellement plus la taille du vocabulaire est grande, causant le problème de malédiction de la dimension (*curse of dimensionality*, en anglais) dû à la rareté des séquences.

Pour alléger ces deux problèmes fondamentaux, (Bengio et al., 2003) propose un modèle de langue neuronal qui utilise un réseau de neurones comme approximateur de la probabilité conditionnelle $p(w_t|w_{t-n+1},\ldots,w_{t-1})$, où chaque symbole du vocabulaire est représenté par un vecteur de nombres réels continus. Comme les données en entrée d'un réseau de neurones doivent être sous format matriciel, chaque symbol w_t se trouvant dans le vocabulaire est représenté par un vecteur *one-hot* $\mathbf{x}_i$ dans lequel toutes les valeurs sont égales à 0, sauf pour la valeur à l'indice i (l'indice du symbole w_t dans le vocabulaire V) qui est égale à 1, tel que $\mathbf{x}_i = [0,0,\ldots,1,\ldots,0,0]^\top \in \{0,1\}^{|V|}$. Ces vecteurs *one-hot* sont multipliés par une matrice de poids, $\mathbf{E} \in \mathbb{R}^{|V|\times d_e}$, qui fait partie des paramètres à apprendre, appelée matrice d'*embeddings*, afin d'obtenir des représentations vectorielles continues, $\mathbf{w}_t \in \mathbb{R}^{d_e}$, qui servent d'entrées au réseau de neurones. Une phrase peut alors être représentée par une séquence de N vecteurs $\mathbf{s} = (\mathbf{w}_1,\ldots,\mathbf{w}_N)$. Nous pouvons voir chaque ligne i de la matrice $\mathbf{E}$ comme la représentation dans un espace vectoriel de dimension d_e correspondant au i-ième symbole du vocabulaire V. Ces représentations sont souvent appelées *word embeddings*. Lorsqu'elles sont apprises sur un corpus assez grand, elles arrivent à représenter la similarité sémantique basée sur la propriété distributionnelle entre les différents symboles du vocabulaire.

L'architecture du modèle de (Bengio et al., 2003) est un réseau de neurones de type *feed-forward* qui définit une séquence de symboles en une distribution de probabilité conditionnelle. Pour l'entraînement des paramètres, nous devons déterminer le nombre des $n-1$ symboles précédents à utiliser pour estimer la probabilité conditionnelle $p(w_t|w_{t-n+1},\ldots,w_{t-1})$. L'entrée du réseau est alors la concaténation d'une séquence de $n-1$ *word embeddings*, appelée le vecteur de contexte $\mathbf{c}$, de sorte que $\mathbf{c} = [\mathbf{w}_{t-n+1}\ ;\ldots\ ;\mathbf{w}_{t-1}]^\top \in \mathbb{R}^{(n-1)d_e}$. Ce vecteur de contexte est acheminé à travers la couche cachée du réseau de neurones qui est une fonction non linéaire d'une transformation affine :

$$\mathbf{h} = tanh(\mathbf{W}_h\mathbf{c} + \mathbf{b}_h), \tag{2.1.3}$$

où $\mathbf{W}_h \in \mathbb{R}^{d_h \times (n-1)d_e}$ et $\mathbf{b}_h \in \mathbb{R}^{d_h}$ sont des paramètres du modèle. Pour calculer la distribution de probabilité de sortie, nous avons besoin d'une fonction qui retourne un vecteur de dimension $|V|$ où chaque élément k est égal à la probabilité conditionnelle estimée $\hat{p}(w_t = k|w_{t-n+1}, \ldots, w_{t-1})$. Une couche de sortie *softmax* garantit des probabilités de sorties positives sommant à 1 :

$$\hat{p}(w_t = k|w_{t-n+1}, \ldots, w_{t-1}) = \frac{\exp(\mathbf{w}_{p,k}^\top \mathbf{h} + b_{p,k})}{\sum_{k' \in V} \exp(\mathbf{w}_{p,k'}^\top \mathbf{h} + b_{p,k'})}, \tag{2.1.4}$$

$$\hat{p}(w_t|w_{t-n+1}, \ldots, w_{t-1}) = softmax(\mathbf{W}_p \mathbf{h} + \mathbf{b}_p), \tag{2.1.5}$$

où $\mathbf{W}_p \in \mathbb{R}^{|V| \times d_h}$ et $\mathbf{b}_p \in \mathbb{R}^{|V|}$ font partie des paramètres du modèle. L'architecture pour ce genre de modèle est présentée à la Figure 2.1.

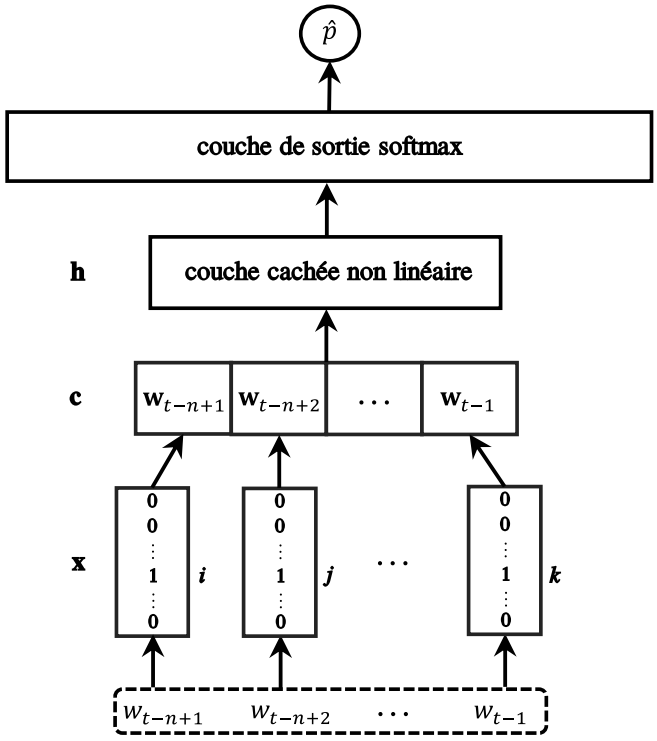


FIGURE 2.1. Architecture d'un modèle de langue neuronal avec un réseau de neurones *feed-forward*.

Il a été démontré que les représentations vectorielles (*word embeddings*) apprises avec des modèles de langue à base de réseaux de neurones sur un ensemble de données suffisamment grand peuvent être utilisées pour améliorer et simplifier de nombreuses applications du TALN, telles que l'étiquetage morpho-syntaxique, l'analyse syntaxique et la reconnaissance d'entités nommées (Collobert and Weston, 2008; Turian et al., 2010). Une extension de ce modèle proposée par (Mikolov et al., 2010) remplace les réseaux de neurones *feed-forward*

par des réseaux de neurones récurrents, qui sont au centre des modèles de langue neuronaux modernes.

2.2. Réseaux de neurones récurrents

Les réseaux de neurones récurrents (RNNs) sont utiles pour traiter les données de tailles variables. Par taille variable, nous voulons dire des données sous la forme de séquences avec un nombre différent d'éléments. Par exemple, deux phrases $\mathbf{s}_i = (w_{i,1}, \ldots, w_{i,N})$ et $\mathbf{s}_j = (w_{j,1}, \ldots, w_{j,M})$ n'ont pas la même longueur si $N \neq M$. La couche cachée d'un RNN, aussi appelée état récurrent, que nous dénotons $\mathbf{h}_t \in \mathbb{R}^{d_h}$, est un vecteur de dimension d_h servant à représenter la mémoire à travers le temps. À chaque pas de temps t, l'état récurrent est une fonction récursive qui se met à jour avec en entrée le symbole actuel w_t et l'état récurrent au pas de temps précédent $\mathbf{h}_{t-1}$. Cette fonction prend la forme :

$$\mathbf{h_t} = f(w_t, \mathbf{h}_{t-1}) \tag{2.2.1}$$

et fait en sorte que la sortie du modèle à chaque pas de temps t est dépendant des calculs précédents.

La fonction $f(\cdot)$ peut prendre plusieurs formes. Dans le cas d'un RNN classique (Elman, 1990), nous avons :

$$f(w_t, \mathbf{h}_{t-1}) = g(\mathbf{W}\psi(w_t) + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}_h), \tag{2.2.2}$$

où $\psi(w_t)$ est une fonction qui transforme un symbole discret en une représentation vectorielle continue de dimension d_e et les matrices $\mathbf{W} \in \mathbb{R}^{d_h \times d_e}$ et $\mathbf{U} \in \mathbb{R}^{d_h \times d_h}$ sont des paramètres du modèle partagés à chaque pas de temps. Ces matrices déterminent l'importance à accorder à la fois au symbol actuel et à l'état récurrent passé. Enfin, $g(\cdot)$ est une fonction d'activation non linéaire. Parmi les plus populaires, nous trouvons les fonctions suivantes :

1. sigmoïde : $\sigma(x) = \frac{1}{1+\exp(-x)}$.
2. tangente hyperbolique : $tanh(x) = \frac{1-\exp(-2x)}{1+\exp(-2x)}$.
3. rectified linear unit : $relu(x) = \max(0, x)$.

La structure du réseau et le format de la sortie y d'un RNN dépendent de la tâche sous-jacente que nous voulons modéliser. Pour les tâches de classification, comme l'extraction de phrases parallèles, l'approche la plus naturelle consiste à utiliser le vecteur du dernier état récurrent afin de produire la sortie désirée (voir le graphique de droite dans la Figure 2.2). Pour modéliser la sortie, la loi de Bernoulli correspond parfaitement à la classification binaire (où la cible $y \in \{0, 1\}$) pour obtenir la probabilité conditionnelle d'avoir un exemple positif :

$$p(y_t = 1|\mathbf{h}_t) = \sigma(\mathbf{v}^\top \mathbf{h}_t + \mathbf{b}_c). \tag{2.2.3}$$

Lorsque nous traitons un problème de classification multiclasse avec n classes, la couche de sortie contient n neurones il est commun d'utiliser une fonction *softmax* pour nous retourner une distribution catégorique, soit la distribution conditionnelle que la sortie $y \in \{1, \ldots, n\}$ définie comme suit :

$$[p(y_t = 1|\mathbf{h}_t), \ldots, p(y_t = n|\mathbf{h}_t)]^\top = softmax(\mathbf{V}^\top \mathbf{h}_t + \mathbf{b}_c), \tag{2.2.4}$$

où $\mathbf{V} \in \mathbb{R}^{d_h \times d_y}$, $\mathbf{v} \in \mathbb{R}^{d_h}$ et $\mathbf{b}_c \in \mathbb{R}^{d_y}$ font partie de l'ensemble des paramètres du modèle.

En théorie, les RNNs permettent de modéliser les dépendances pour des séquences arbitrairement longues. En réalité, il est démontré qu'il est difficile d'apprendre des dépendances à long terme durant l'entraînement des paramètres (Bengio et al., 1994). Comme un RNN est récursif et que les paramètres sont partagés à travers chaque pas de temps pour une séquence en entrée, en appliquant l'algorithme de rétropropagation du gradient pour l'apprentissage des paramètres du modèle, le gradient ∇ dépend non seulement des calculs au pas de temps actuel, mais aussi aux pas de temps précédents. Pour une séquence de N symboles, cela peut avoir comme effet de multiplier N petits nombres pour calculer le gradient, de sorte que sa valeur décroît exponentiellement pour tendre rapidement vers zéro. Cela a comme effet que le modèle apprend très lentement après quelques pas de temps, voir pas du tout. Cet effet problématique est appelé le *vanishing gradient problem*. De l'autre côté, lorsque nous avons un gradient qui peut prendre des valeurs élevées, nous faisons face au *exploding gradient problem*.

Le problème du *exploding gradient* est relativement facile à corriger. Nous pouvons le détecter en inspectant la norme du gradient de la fonction de coût à optimiser par rapport aux paramètres du modèle, dénotée $\|\nabla\|$. Une solution simple est proposée par (Pascanu et al., 2013) où il s'agit de normaliser la norme du gradient si elle est supérieure à un seuil prédéfini τ :

$$\tilde{\nabla} = \begin{cases} \tau \frac{\nabla}{\|\nabla\|}, & \text{si } \|\nabla\| > \tau, \\ \nabla, & \text{sinon.} \end{cases} \tag{2.2.5}$$

En revanche, il n'y a pas de moyen simple pour éviter le problème du *vanishing gradient*. Une solution populaire, mais pas parfaite, consiste à utiliser des architectures *Long Short Term Memory* (LSTM) ou *Gated Recurrent Unit* (GRU). Ces architectures sont couvertes dans les deux sections suivantes.

2.2.1. Long short term memory

L'architecture *Long Short Term Memory* a d'abord été proposée par (Hochreiter and Schmidhuber, 1997) dans le but de construire un RNN capable d'apprendre les dépendances à long terme sur un grand nombre de pas de temps. Aujourd'hui, elle est l'architecture la plus

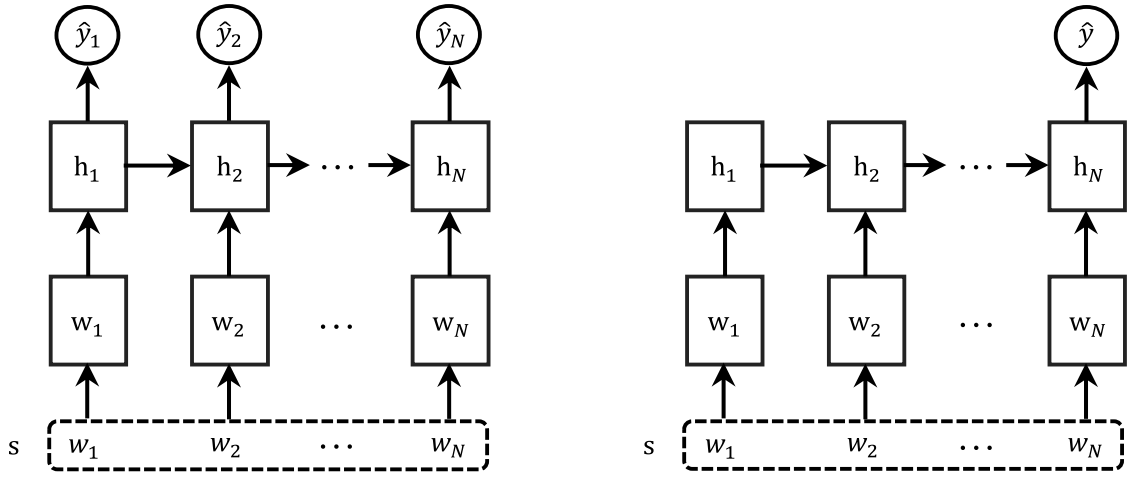


FIGURE 2.2. Deux exemples de graphes de réseaux de neurones récurrents. À gauche, le RNN a une sortie à chaque pas de temps. À droite, une seule sortie a lieu lorsque le RNN a traité la séquence en entier jusqu'à la fin.

utilisée dans les applications d'apprentissage profond en TALN. Un LSTM n'a pas une architecture fondamentalement différente d'un RNN, mais il utilise une fonction différente et plus sophistiquée dans le calcul des états récurrents (Figure 2.3).

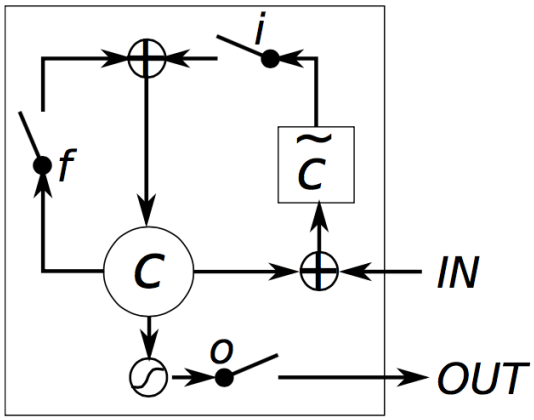


FIGURE 2.3. Illustration du fonctionnement d'un LSTM.

Contrairement à un RNN classique, un LSTM sépare la mémoire en deux composantes ; l'état de mémoire $\mathbf{c_t}$ et l'état récurrent $\mathbf{h_t}$. L'état récurrent $\mathbf{h_t}$ est un sous-ensemble de l'état de mémoire cachée $\mathbf{c_t}$ et seulement ce sous-ensemble est visiblement exposé aux autres parties du réseau. Le LSTM utilise une porte de sortie $\mathbf{o}$ (*output gate*, en anglais) pour déterminer combien de mémoire révéler à l'état récurrent. La fonction de l'*output gate* est calculée ainsi :

$$\mathbf{o} = \sigma(\mathbf{W}_o \psi(w_t) + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o). \tag{2.2.6}$$

Ce vecteur est multiplié élément par élement avec l'état de mémoire pour donner l'état récurrent suivant :

$$\mathbf{h}_t = \mathbf{o} \odot tanh(\mathbf{c}_t). \tag{2.2.7}$$

Pour mettre à jour l'état de mémoire, le LSTM utilise une porte d'oublie $\mathbf{f}$ (*forget gate*, en anglais) et une porte d'entrée $\mathbf{i}$ (*input gate*, en anglais), de sorte que :

$$\mathbf{c}_t = \mathbf{f} \odot \mathbf{c}_{t-1} + \mathbf{i} \odot \tilde{\mathbf{c}}_t, \tag{2.2.8}$$

où $\tilde{\mathbf{c}}_t$ est un état de mémoire candidat. La *forget gate* détermine la proportion d'information à oublier de l'état de mémoire, tandis que la *input gate* contrôle la proportion d'information à garder du nouvel élément en entrée et de l'état récurrent précédent :

$$\mathbf{f} = \sigma(\mathbf{W}_f \psi(w_t) + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \tag{2.2.9}$$

$$\mathbf{i} = \sigma(\mathbf{W}_i \psi(w_t) + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \tag{2.2.10}$$

$$\tilde{\mathbf{c}} = tanh(\mathbf{W}_c \psi(w_t) + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c). \tag{2.2.11}$$

Les paramètres $\mathbf{W}_o$, $\mathbf{U}_o$, $\mathbf{b}_o$, $\mathbf{W}_f$, $\mathbf{U}_f$, $\mathbf{b}_f$, $\mathbf{W}_i$, $\mathbf{U}_i$, $\mathbf{b}_i$, $\mathbf{W}_c$, $\mathbf{U}_c$ et $\mathbf{b}_c$ sont des paramètres supplémentaires spécifiques au LSTM et doivent être estimés avec tous les autres paramètres du modèle. Plusieurs variantes mineures ont été proposées au LSTM à travers le temps (Greff et al., 2015; Jozefowicz et al., 2015).

2.2.2. Gated recurrent unit

Le *Gated Recurrent Unit* a été introduit par (Cho et al., 2014) avec l'idée d'apprendre les dépendances à long terme d'une séquence en utilisant un méchanisme de portes proche de celui du LSTM. Contrairement au LSTM, la mémoire du modèle n'est pas séparée en deux composantes et elle est complètement représentée dans les état récurrents $\mathbf{h}_t$ (comme un RNN classique). La fonction de calcul de la mise à jour des états récurrents du GRU consiste en un méchanisme de portes permettant de laisser passer une proportion d'information, sauf qu'elle en possède deux au lieu de trois pour un LSTM (voir Figure 2.4). La porte de réinitialisation $\mathbf{r}$ (*reset gate*, en anglais) détermine la proportion d'information des états récurrents précédents qui sera utilisée pour calculer l'état récurrent candidat $\tilde{\mathbf{h}}_t$, tel que :

$$\mathbf{r} = \sigma(\mathbf{W}_r \psi(w_t) + \mathbf{U}_r \mathbf{h}_{t-1}), \tag{2.2.12}$$

$$\tilde{\mathbf{h}}_t = tanh(\mathbf{W}_h \psi(w_t) + \mathbf{U}_h(\mathbf{r} \odot \mathbf{h}_{t-1})). \tag{2.2.13}$$

Par la suite, une porte de mise à jour $\mathbf{u}$ (*update gate*, en anglais) contrôle le niveau d'information à mettre à jour dans le nouvel état récurrent, c'est-à-dire qu'elle détermine à chaque pas de temps t la proportion d'information de l'état récurrent candidat $\tilde{\mathbf{h}}_t$ à utiliser pour calculer le nouvel état récurrent $\mathbf{h}_t$:

$$\mathbf{u} = \sigma(\mathbf{W}_u \psi(w_t) + \mathbf{U}_u \mathbf{h}_{t-1}), \tag{2.2.14}$$

$$\mathbf{h}_t = (1 - \mathbf{u}) \odot \mathbf{h}_{t-1} + \mathbf{u} \odot \tilde{\mathbf{h}}_t. \tag{2.2.15}$$

Les paramètres $\mathbf{W}_r$, $\mathbf{U}_r$, $\mathbf{W}_u$ et $\mathbf{U}_u$ sont des paramètres supplémentaires spécifiques au GRU.

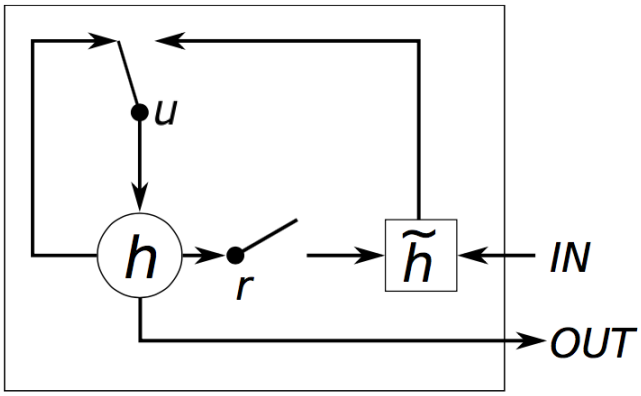


FIGURE 2.4. Illustration du fonctionnement d'un GRU.

2.2.3. Réseaux de neurones récurrents bidirectionnels

Les réseaux de neurones récurrents bidirectionnels (BiRNNs) (Schuster and Paliwal, 1997) sont une extension des RNNs basée sur l'idée que la sortie au pas de temps t peut non seulement dépendre des éléments précédents de la séquence, mais aussi des éléments futurs. Par exemple, lors de l'apprentissage d'un modèle de traduction automatique neuronal, il peut être avantageux d'utiliser l'information du contexte qui se trouve des deux côtés du symbole actuel. Du point de vue de l'architecture, un BiRNN est simplement deux RNNs empilés l'un sur l'autre. Le RNN *forward* est un RNN qui traite les éléments un à la fois du début jusqu'à la fin de la séquence (1 à N), alors que le RNN *backward* traite les éléments de la fin jusqu'au début (N à 1). Nous définissons les états récurrents au pas de temps t des RNNs *forward* et *backward* $\overrightarrow{\mathbf{h}}_t$ et $\overleftarrow{\mathbf{h}}_t$, respectivement. Diverses techniques sont utilisées pour combiner les représentations vectorielles $\overrightarrow{\mathbf{h}}_t$ et $\overleftarrow{\mathbf{h}}_t$ dans le but d'obtenir une seule représentation vectorielle de la séquence. Les trois plus populaires sont définies ci-dessous :

1. *Concatenate* : concaténisation des derniers états récurrents de chaque RNN, $\mathbf{h} = \left[\overrightarrow{\mathbf{h}}_N \,;\, \overleftarrow{\mathbf{h}}_1\right]$.
2. *Mean pooling* : moyenne des états récurrents concaténisés à chaque pas de temps, $\mathbf{h} = \frac{1}{N} \sum_{t=1}^{N} \left[\overrightarrow{\mathbf{h}}_t \,;\, \overleftarrow{\mathbf{h}}_t\right]$.
3. *Max pooling* : maximum sur chaque dimension de tous les états récurrents concaténisés à chaque pas de temps, $\mathbf{h} = \max\left(\left[\left[\overrightarrow{\mathbf{h}}_t \,;\, \overleftarrow{\mathbf{h}}_t\right]\right]_{t=1}^{N}\right)$.

2.2.4. Méthodes d'apprentissage

Aux sections précédentes, nous avons décrit différentes architectures d'apprentissage profond nous permettant de modéliser des séquences de tailles variables. Dans cette section, nous présentons comment faire l'apprentissage des paramètres de ces modèles de manière efficace.

2.2.4.1. *Algorithme du gradient*

L'algorithme d'optimisation le plus utilisé pour estimer l'ensemble des paramètres θ qui minimise une fonction de coût $J(\theta)$ est l'algorithme du gradient (*gradient descent*, en anglais). Une fonction de perte couramment utilisée est la fonction de log-vraisemblance négative (*negative log-likelihood*, en anglais) sur les données d'entraînement. L'algorithme du gradient est une méthode d'optimisation itérative qui modifie l'ensemble des paramètres successivement jusqu'à ce que la fonction de coût sur les données d'entraînement converge vers la solution optimale. À chaque pas de temps, nous avançons d'un pas dans la direction opposée au gradient de la fonction de coût par rapport aux paramètres du modèle pour les mettre à jour de sorte que $J(\theta)$ se dirige vers un minimum local [1] :

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta), \tag{2.2.16}$$

où α est le taux d'apprentissage (*learning rate*, en anglais), un hyperparamètre important pour réussir à faire un bon apprentissage. En effet, il est crucial de bien ajuster le taux d'apprentissage α à l'initialisation et durant l'optimisation des paramètres. Une valeur trop faible fait en sorte que la fonction de coût converge très lentement vers un minimum local. D'un autre côté, si la valeur est trop élevée l'optimisation risque de faire un pas d'une distance trop grande et de ne jamais réussir à atteindre un minimal local.

Avec la croissance de la taille des ensembles de données d'entraînement utilisés pour faire l'apprentissage des modèles, il est devenu de plus en plus computationnellement exigeant de calculer $\nabla J(\theta)$. Ce coût de calcul associé à chaque itération du processus d'optimisation a motivé l'utilisation de l'algorithme du gradient stochastique (SGD) (*stochastic gradient descent*, en anglais). Au lieu de calculer à chaque itération le gradient sur toutes les données de l'ensemble d'entraînement, SGD utilise un sous-ensemble des données, appelé mini-batch, qui est choisi au hasard pour calculer l'estimation du gradient pour mettre à jour la valeur des paramètres. En plus de rendre l'optimisation plus rapide, il a été démontré qu'utiliser des mini-batches permet de converger vers une meilleure solution pour les problèmes non convexes de grande dimension, comme c'est le cas avec les réseaux de neurones récurrents (LeCun et al., 1998). Habituellement, nous visitons les données de l'ensemble d'entraînement plusieurs fois et chaque passage est appelé une époque (*epoch*, en

1. Vers un minimum global si la fonction de coût est convexe, ce qui n'est pas le cas pour la très grande majorité des problèmes où l'apprentissage profond est appliqué.

anglais).

Dû au grand nombre de paramètres à estimer et à la complexité des architectures des réseaux de neurones modernes qui consiste en une composition de plusieurs sous-fonctions, le calcul du gradient n'est souvent pas trivial. (Rumelhart et al., 1986) a introduit ce que nous appelons maintenant l'algorithme de rétropropagation du gradient qui est un moyen efficace de calculer le gradient des fonctions.

2.2.4.2. *Adam*

Depuis la montée de l'apprentissage profond, plusieurs variantes par rapport à SGD ont été développées pour accélérer ou améliorer le processus d'apprentissage. Notre objectif n'étant pas de faire un résumé exhaustif de ces algorithmes d'optimisation, nous présentons seulement celle que nous utilisons dans nos expériences pour entraîner les modèles de notre approche. Une comparaison approfondie de ces méthodes d'apprentissage est faite dans (Ruder, 2016).

L'algorithme d'optimisation que nous utilisons est Adam (Kingma and Ba, 2014). Adam calcule des taux d'apprentissages adaptatifs pour chaque paramètre. Le calcul de la mise à jour des paramètres se fait comme suit :

$$g_t = \nabla J(\theta_{t-1}), \tag{2.2.17}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \tag{2.2.18}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2, \tag{2.2.19}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \tag{2.2.20}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}, \tag{2.2.21}$$

$$\alpha_t = \alpha \frac{\sqrt{(1-\beta_2^t)}}{(1-\beta_1^t)}, \tag{2.2.22}$$

$$\theta_t = \theta_{t-1} - \alpha_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{2.2.23}$$

où α est le taux d'apprentissage initial, β_1 et β_2 sont des hyperparamètres $\in [0, 1)$ qui contrôlent le taux de décroissance exponentielle des moyennes mobiles du gradient m_t et du carré du gradient v_t. Ces moyennes mobiles sont des estimations de la moyenne et de la variance non centrée du gradient. Les auteurs recommandent d'initialiser les hyperparamètres aux valeurs suivantes ; $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ et $\epsilon = 10^{-8}$.

2.3. Représentations vectorielles multilingues

Suite à la popularité fulgurante des propriétés sémantiques des *word embeddings* entraînés sur de larges corpus monolingues, une prochaine étape qui est venue naturellement à l'esprit des chercheurs est d'apprendre des représentations dans un espace vectoriel partagé où les mots dans plusieurs langues sont regroupés. Dans un scénario optimal, un tel espace vectoriel nous permettrait d'entraîner et d'utiliser un modèle dans toutes les langues.

En effet, (Mikolov et al., 2013b) démontre que les espaces vectoriels des représentations des mots présentent des structures similaires à travers les langues, même en considérant une paire de langue distante comme l'anglais et le vietnamien. Les auteurs proposent de mettre en correspondance deux espaces vectoriels monolingues en appliquant une transformation linéaire afin de projeter les représentations de l'espace vectoriel d'une langue source $\mathbf{w}^s$ vers les représentations de l'espace vectoriel d'une langue cible $\mathbf{w}^c$. L'objectif est alors d'utiliser les représentations vectorielles de n paires de mots bilingues et d'apprendre une matrice de projection $\mathbf{W} \in \mathbb{R}^{d_c \times d_s}$ où d_s et d_c sont respectivement la dimension des représentations vectorielles des langues source et cible, de manière à ce que :

$$\min_{W} \sum_{i=1}^{n} \|\mathbf{W}\mathbf{w}_i^s - \mathbf{w}_i^c\| \,. \tag{2.3.1}$$

Une illustration d'un exemple d'espace vectoriel partagé entre des mots en anglais et en allemand adapté de (Luong et al., 2015) est présentée à la Figure 2.5.

Depuis, plusieurs travaux visant à améliorer l'estimation des représentations vectorielles multilingues ont découlé de cette approche. Par exemple, (Gouws et al., 2015) propose d'ajouter un terme de régularisation interlangue Ω lors de l'apprentissage des représentations vectorielles

$$\Omega_t = \left\| \frac{1}{N} \sum_{w_i \in \mathbf{s}^s}^{N} \mathbf{w}_i^s - \frac{1}{M} \sum_{w_j \in \mathbf{s}^c}^{M} \mathbf{w}_i^c \right\|, \tag{2.3.2}$$

où $\mathbf{s}^s$ et $\mathbf{s}^c$ sont respectivement une phrase source de longueur N et une phrase cible de longueur M. Cette fonction aligne uniformément tous les mots qui sont présents dans une paire de phrase parallèle en minimisant la distance entre la moyenne de la représentation vectorielle des mots dans chaque phrase. Une étude approfondie de ces approches est faite par (Ruder, 2017).

Même si une vaste collection de modèles a été proposée depuis les dernières années, ces modèles doivent utiliser des données parallèles comme des lexiques bilingues, des corpus parallèles ou des documents alignés pour réussir à aligner les mots dans plusieurs langues dans un espace vectoriel commun. Comme la quantité de ces données parallèles est limitée pour la

grande majorité des paires de langues, le développement et l'applicabilité de ces modèles reste un enjeu principal. Toutefois, un récent travail (Conneau et al., 2017) indique le contraire. Dans ce travail, les auteurs montrent qu'il est possible d'induire un lexique bilingue entre deux langues en alignant les espaces vectoriels monolingues de manière non supervisée sans l'utilisation de données parallèles. Les auteurs proposent d'utiliser l'apprentissage accusatoire (*adversarial training*, en anglais) pour apprendre une matrice de paramètres qui sert à projeter les représentations de l'espace vectoriel de la langue source vers celui de la langue cible.

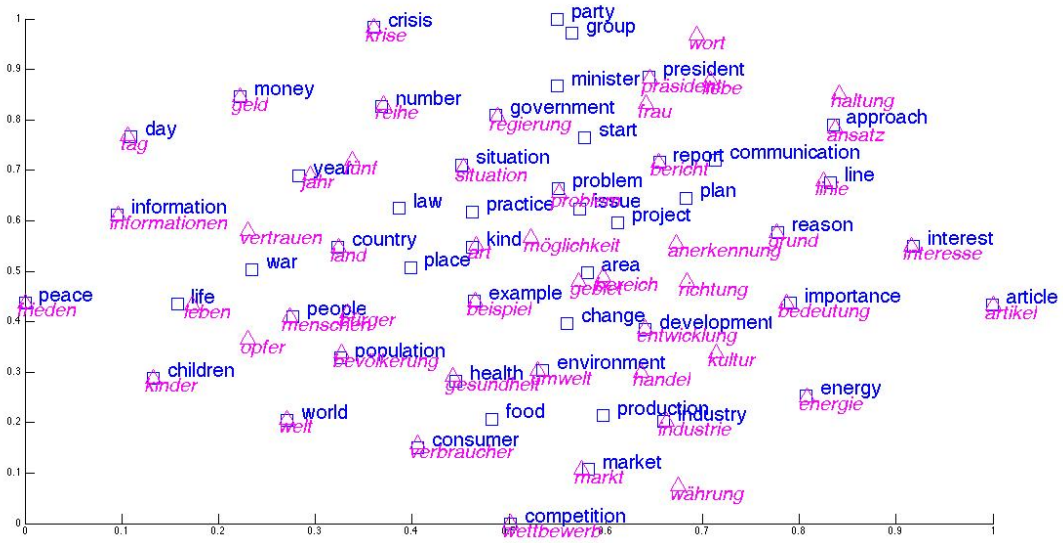


FIGURE 2.5. Espace vectoriel partagant des mots en anglais et en allemand (Luong et al. (2015)).

Chapitre 3

APPROCHE PROPOSÉE

Dans ce chapitre nous présentons l'approche que nous proposons pour classifier et extraire des phrases potentiellement en relation de traduction. Cette approche s'applique sur n'importe quelle sorte de paire de textes dans deux langues. L'idée que nous voulons tester est d'étendre les systèmes décrits au chapitre 1 en réalisant cette tâche à partir des méthodes présentées au chapitre 2. En premier lieu, nous voulons utiliser les réseaux de neurones récurrents pour encoder les phrases en représentations vectorielles afin de tirer profit des propriétés distributionnelles de ces représentations. Deuxièmement, dans la même lignée que les modèles à représentations vectorielles multilingues, nous voulons apprendre les représentations dans un espace vectoriel partagé. Dans un tel espace, nous cherchons à ce que la distance entre les représentations vectorielles des phrases parallèles soit petite et que les phrases non parallèles soient de plus en plus éloignées en fonction de leur degré de similarité.

3.1. Sélection d'exemples négatifs

Avant de présenter l'architecture du modèle, nous devons préciser comment nous sélectionnons les exemples négatifs, c'est-à-dire les paires de phrases non parallèles. Nous rappelons que notre objectif est d'avoir un classificateur qui nous donne la probabilité qu'une paire de phrases dans deux langues soient en relation de traduction. Pour y arriver, nous devons entraîner ce classificateur sur un ensemble d'entraînement constitué d'exemples positifs et négatifs de manière à ce qu'il puisse différencier les phrases parallèles des nombreuses phrases non parallèles. Obtenir une frontière de décision parfaite en utilisant uniquement des paires de phrases comme source de données d'entraînement n'est pas réaliste étant donné que deux phrases parfaitement traduites peuvent avoir plusieurs traductions correctes. C'est pourquoi nous avons besoin d'une fonction qui renvoie une distribution de probabilité qui attribue des probabilités élevées aux traductions probables. De plus, lors de l'étape d'extraction, les paires de phrases retrouvées dans des documents comparables peuvent avoir différents degrés de similarité. Par exemple, il est fort probable de trouver des phrases étroitement reliées dans lesquelles seulement quelques mots sont manquants. Nous voulons

donc avoir un modèle assez flexible pour extraire ces paires de phrases qui ajoutent de la valeur à un modèle de traduction automatique même si elles ne sont pas parfaitement parallèles.

Comme exemples positifs pour notre ensemble d'entraînement, nous utilisons un corpus parallèle C contenant n paires de phrases parallèles $\{(\mathbf{s}_k^s, \mathbf{s}_k^c)\}_{k=1}^n$, où $\mathbf{s}_k^s$ et $\mathbf{s}_k^c$ dénotent respectivement des phrases de l'ensemble des phrases des langues source et cible. Puisque nous voulons un modèle qui apprend des représentations vectorielles différentiables pour distinguer les phrases parallèles des phrases non parallèles, nous utilisons l'échantillonnage négatif aléatoire pour générer des exemples négatifs. Pour y arriver, pour chaque phrase source positive nous échantillonnons m phrases cibles négatives de manière à créer m nouveaux exemples négatifs, tel que $(\mathbf{s}_k^s, \mathbf{s}_j^c)$ avec $j \neq k$. Ce processus est répété à nouveau au début de chaque époque d'entraînement pour permettre au modèle d'apprendre sur un plus grand ensemble de paires non parallèles. Cela dit, à chaque époque d'entraînement notre ensemble d'entraînement est composé d'un total de $n(1+m)$ triplets $(\mathbf{s}_i^s, \mathbf{s}_i^c, y_i)$, où $\mathbf{s}_i^s = (w_{i,1}^s, \ldots, w_{i,N}^s)$ est une phrase dans la langue source de N symboles, $\mathbf{s}_i^c = (w_{i,1}^c, \ldots, w_{i,M}^c)$ est une phrase dans la langue cible de M symboles, et y_i est une cible binaire représentant la relation de traduction entre $\mathbf{s}_i^s$ et $\mathbf{s}_i^c$, telle que $y_i = 1$ si $(\mathbf{s}_i^s, \mathbf{s}_i^c) \in C$, sinon $y_i = 0$.

L'avantage de l'échantillonnage négatif aléatoire est sa simplicité. D'une certaine manière, avec notre modèle nous faisons l'hypothèse sous-jacente que les paires de mots se trouvant dans les paires de phrases parallèles devraient être rapprochées dans l'espace vectoriel, alors que tous les autres mots qui se trouvent dans les paires de phrases non parallèles devraient se distancer.

3.2. Modèle

Notre idée est d'utiliser les modèles à base de réseaux de neurones profonds pour apprendre la similarité entre des paires de phrases $(\mathbf{s}_i^s, \mathbf{s}_i^c)$ afin d'estimer la probabilité qu'elles soient des traductions l'une de l'autre :

$$p(y_i = 1 | \mathbf{s}_i^s, \mathbf{s}_i^c). \tag{3.2.1}$$

L'architecture du modèle proposé est un réseau siamois (Bromley et al., 1993) constitué de réseaux de neurones récurrents bidirectionnels (voir section 2.2.3). Un BiRNN agit en tant qu'encodeur de phrases, c'est-à-dire qu'il convertit des séquences de symboles de tailles variables en des représentations vectorielles de dimensions fixes. Le BiRNN peut utiliser n'importe quelle des fonctions d'activations récurrentes pour mieux modéliser les effets de dépendance à long terme, telles que les fonctions LSTM (voir section 2.2.1) et GRU (voir section 2.2.2). Puisque nous voulons comparer des paires de phrases, un réseau siamois avec des poids partagés nous permet d'encoder deux phrases en des représentations vectorielles

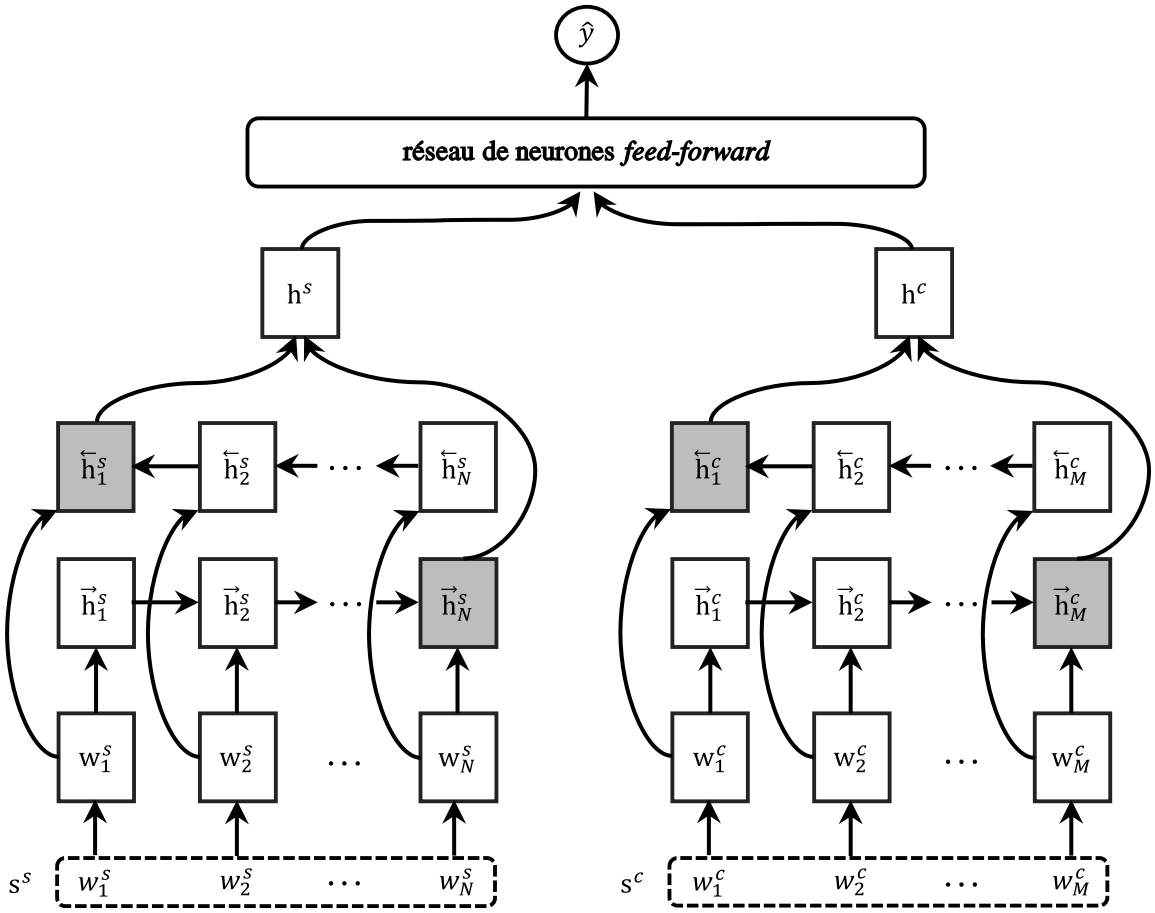


FIGURE 3.1. Architecture du réseau de neurones récurrents bidirectionnels siamois.

dans un espace vectoriel partagé. Un réseau siamois avec des poids partagés signifie que les deux BiRNNs partagent les mêmes valeurs pour leur ensemble de paramètres, ce qui est équivalent à utiliser le même BiRNN pour encoder les deux phrases de chaque paire en deux représentations vectorielles. Pour chaque phrase, nous concaténons les derniers états récurrents des RNNs *forward* et *backward* pour obtenir sa représentation vectorielle. Ces représentations sont ensuite acheminées dans un réseau de neurones *feed-forward* avec une couche de sortie sigmoïde qui nous permet de mesurer la probabilité qu'elles soient parallèles. L'architecture de notre modèle est présentée à la Figure 3.1.

Ci-dessous nous définissons les équations du modèle pour l'encodage d'une phrase source. Pour une phrase cible, il suffit de remplacer s pour c dans les équations. À chaque pas de temps t, le symbole de la i-ième phrase $w_{i,t}^s$ défini par le nombre entier représentant son indice dans le vocabulaire V^s, est représenté par un vecteur *one-hot*, $\mathbf{x}_k^s \in \{0,1\}^{|V|}$,

dont le k-ième élément est égal à 1 et tous les autres éléments à 0. Le produit matriciel entre la matrice d'*embeddings* du langage source, $\mathbf{E}^s \in \mathbb{R}^{|V^s| \times d_e}$, avec ce vecteur *one-hot* est appliqué afin d'obtenir la représentation vectorielle continue de ce symbole, $\mathbf{w}^s_{i,t} \in \mathbb{R}^{d_e}$. Cette représentation vectorielle sert d'entrée pour les états récurrents de l'encodeur BiRNN, où $\overrightarrow{\mathbf{h}}^s_{i,t} \in \mathbb{R}^{d_h}$ et $\overleftarrow{\mathbf{h}}^s_{i,t} \in \mathbb{R}^{d_h}$ sont respectivement les états récurrents des RNNs *forward* et *backward*. Nous devons fixer d_e et d_h, qui dénotent respectivement les dimensions des représentations vectorielles des symboles et des états récurrents. Pour chaque phrase, le RNN *forward* procède un symbole à la fois en mettant à jour l'état récurrent à partir du premier symbole jusqu'au dernier pour obtenir une représentation vectorielle de taille fixe $\overrightarrow{\mathbf{h}}^s_{i,N}$. Le RNN *backward* traite la phrase en sens inverse, c'est-à-dire qu'il procède la phrase à partir du dernier symbole jusqu'au premier en mettant à jour l'état récurrent pour obtenir une représentation vectorielle de taille fixe $\overleftarrow{\mathbf{h}}^s_{i,1}$. Nous faisons la concaténation du dernier état récurrent dans les deux directions comme représentation vectorielle finale (voir Figure 3.1). Pour obtenir une représentation vectorielle finale, nous avons considéré de combiner les états récurrents avec les opérations *mean pooling* et *max pooling* (voir section 2.2.3), mais nous avons obtenu des performances inférieures.

Les étapes que nous venons de décrire pour encoder une phrase en représentation vectorielle continue peuvent être définies comme :

$$\mathbf{w}^s_{i,t} = \mathbf{E}^{s^\top} \mathbf{x}^s_k, \tag{3.2.2}$$

$$\overrightarrow{\mathbf{h}}^s_{i,t} = f(\mathbf{w}^s_{i,t}, \overrightarrow{\mathbf{h}}^s_{i,t-1}), \tag{3.2.3}$$

$$\overleftarrow{\mathbf{h}}^s_{i,t} = f(\mathbf{w}^s_{i,t}, \overleftarrow{\mathbf{h}}^s_{i,t+1}), \tag{3.2.4}$$

$$\mathbf{h}^s_i = \left[\overrightarrow{\mathbf{h}}^s_{i,N} ; \overleftarrow{\mathbf{h}}^s_{i,1}\right], \tag{3.2.5}$$

où $f(\cdot)$ peut être n'importe quelle fonction d'activation récurrente, telle qu'un LSTM ou un GRU.

Une fois que les phrases source et cible ont été encodées, nous les alimentons dans un réseau de neurones *feed-forward* pour calculer la probabilité qu'elles soient une traduction l'une de l'autre. Tout d'abord, nous mesurons leur similarité en utilisant la multiplication par élément et la différence absolue pour mesurer l'angle et la distance entre ces deux représentations vectorielles. Nous combinons ces deux nouvelles représentations vectorielles en appliquant une transformation affine et une fonction non linéaire. Finalement le résultat de cette dernière représentation culmine vers une couche de sortie sigmoïde pour calculer la

probabilité conditionnelle d'avoir un exemple positif en sortie :

$$\mathbf{h}_i^{(1)} = \mathbf{h}_i^s \odot \mathbf{h}_i^c, \tag{3.2.6}$$

$$\mathbf{h}_i^{(2)} = |\mathbf{h}_i^s - \mathbf{h}_i^c|, \tag{3.2.7}$$

$$\mathbf{h}_i = tanh(\mathbf{W}^{(1)}\mathbf{h}_i^{(1)} + \mathbf{W}^{(2)}\mathbf{h}_i^{(2)} + \mathbf{b}), \tag{3.2.8}$$

$$p(y_i = 1|\mathbf{h}_i) = \sigma(\mathbf{v}\mathbf{h}_i + b). \tag{3.2.9}$$

$\sigma(\cdot)$ est la fonction sigmoïde, $\mathbf{W}^{(1)} \in \mathbb{R}^{d_f \times d_h}$, $\mathbf{W}^{(2)} \in \mathbb{R}^{d_f \times d_h}$, $\mathbf{v} \in \mathbb{R}^{d_f}$, $\mathbf{b} \in \mathbb{R}^{d_f}$ et b sont des paramètres du modèle. La valeur d_f est la dimension de la couche cachée du réseau *feed-forward*.

Le modèle est entraîné en minimisant l'entropie croisée des paires de phrases de notre ensemble d'entraînement :

$$\begin{aligned} \mathcal{L} &= -\sum_{i=1}^{n(1+m)} \log p(y_i|\mathbf{h}_i), \\ \mathcal{L} &= -\sum_{i=1}^{n(1+m)} y_i \log \sigma(\mathbf{v}\mathbf{h}_i + b) - (1 - y_i) \log(1 - \sigma(\mathbf{v}\mathbf{h}_i + b)). \end{aligned} \tag{3.2.10}$$

Au moment de faire des prédictions une fois que le modèle est entraîné, une paire de phrases est classée comme parallèle si la probabilité obtenue est supérieure ou égale à un seuil de décision ρ que nous devons fixer :

$$\hat{y}_i = \begin{cases} 1 & \text{si } p(y_i = 1|\mathbf{h}_i) \geq \rho, \\ 0 & \text{sinon.} \end{cases} \tag{3.2.11}$$

Les deux matrices d'*embeddings*, $\mathbf{E}^s$ et $\mathbf{E}^c$, sont des paramètres du modèle que nous devons apprendre. Plus la taille du vocabulaire et la dimension d_e augmentent, plus le nombre de paramètres peut devenir considérablement coûteux à estimer. C'est pourquoi il est commun d'initialiser une matrice d'*embeddings* en utilisant des *word embeddings* pré-entraînés sur une large collection de textes. Pour de nombreuses applications monolingues du TALN, plusieurs *word embeddings* pré-entraînés de bonne qualité sont offerts à la communauté. Parmi les représentations vectorielles pré-entraînées les plus populaires, nous trouvons celles entraînées par les applications word2vec (Mikolov et al., 2013a) [1], GloVe (Pennington et al., 2014) [2] et fastText (Bojanowski et al., 2016) [3]. Toutes ces représentations vectorielles sont représentées dans un espace vectoriel $\mathbb{R}^{300}$. Malheureusement, il n'existe pas de représentations vectorielles multilingues d'aussi bonne qualité offertes pour les applications multilingues.

1. https ://code.google.com/archive/p/word2vec/
2. https ://nlp.stanford.edu/projects/glove/
3. https ://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

C'est encore possible d'entraîner nos propres représentations vectorielles multilingues dans un espace conjoint en utilisant un des nombreux modèles de ce domaine (Mikolov et al., 2013b; Gouws et al., 2015; Smith et al., 2017) et de les utiliser pour initialiser les matrices d'*embeddings*. Bien qu'il soit possible d'initialiser ces matrices à partir de représentations vectorielles pré-entraînés afin d'accélérer l'apprentissage et même d'améliorer les performances de notre modèle, nous préférons les apprendre à partir de zéro et de ne pas avoir à utiliser des ressources externes supplémentaires.

Chapitre 4

EXPÉRIENCES ET RÉSULTATS

Pour évaluer l'efficacité de notre approche, nous la testons dans différents contextes. Tout d'abord, dans la section 4.1 nous mesurons sa capacité à identifier les phrases parallèles retrouvées dans des corpus parallèles standards. Pour comparer les systèmes sur des corpus pseudo-comparables avec différents degrés de comparabilité, nous ajoutons du bruit en insérant des phrases non parallèles dans ces corpus parallèles. Dans la section 4.2, nous extrayons des paires de phrases d'un corpus comparable et validons leur utilité en mesurant leur impact sur des systèmes de traduction automatique SMT et NMT.

4.1. Évaluation intrinsèque des modèles

4.1.1. Données

La façon la plus fiable de comparer notre approche avec un système de référence serait de les évaluer sur les phrases d'une collection de textes où des traducteurs professionnels ont annoté manuellement les phrases parallèles. Cependant, cette option est coûteuse et peu pratique. Par conséquent, pour la tâche d'extraction de phrases parallèles il est courant de comparer différentes approches en utilisant des textes alignés à partir d'un corpus parallèle standard. Pour calculer nos critères d'évaluation, nous utilisons des ensembles de données anglais et français de WMT'15 [1]. WMT, *Workshop on Statistical Machine Translation*, est une conférence annuelle très populaire dans le domaine de la traduction automatique. Les données partagées pour les tâches de traduction de WMT sont souvent utilisées dans la recherche et la pratique pour entraîner et évaluer des systèmes utilisant des corpus parallèles comme ressources. Les exemples positifs de notre ensemble d'entraînement sont 500k paires de phrases parallèles que nous avons échantillonées au hasard du corpus Europarl. Pour générer les exemples négatifs, au début de chaque époque nous échantillonnons m phrases non parallèles par exemple positif, tel que décrit dans la section 3.1. Le nombre total de paires de phrases dans l'ensemble d'entraînement est donc de 500,000$(m + 1)$. La taille du

1. http ://www.statmt.org/wmt15/translation-task.html

vocabulaire est de 69k pour l'anglais et 84k pour le français.

Nous soutenons qu'un bon système d'extraction de phrases parallèles dans la pratique doit idéalement généraliser sur d'autres domaines que celui présent dans l'ensemble d'entraînement. En d'autres mots, les données au moment de faire la prédiction qu'une paire de phrases soit parallèle ou non couvriront très probablement d'autres domaines que ceux trouvés dans l'ensemble des données utilisé pour faire l'apprentissage du système. Par conséquent, le cadre de notre protocole expérimental répond à cette situation en utilisant les 1,000 premières paires de phrases parallèles du corpus newstest2012 pour évaluer nos modèles. Afin d'évaluer la robustesse de notre approche sur différents degrés de comparabilité de corpus comparables, pour créer nos ensembles de test nous introduisons des phrases non parallèles en substituant un nombre défini de phrases françaises par d'autres phrases françaises du corpus newstest2012 qui ne font pas partie de l'ensemble des 1,000 phrases sélectionnées. Cela nous permet de réduire le degré de parallélisme en augmentant le nombre de paires de phrases non parallèles dans nos ensembles de test. Nous avons donc 10 ensembles de test basés sur newstest2012 avec des ratios de bruit de $\{0, 0.1, \ldots, 0.8, 0.9\}$.

Nous évaluons quand même la capacité d'un système à classifier des paires de phrases venant du même domaine, mais qui n'ont pas été observées durant son entraînement. Nous utilisons 3 ensembles de test avec des ratios de bruit de $\{0, 0.5, 0.9\}$ à partir de 1,000 paires de phrases échantillonnées du corpus Europarl, en nous assurant qu'elles ne se trouvent pas dans les ensembles d'entraînement.

Chaque ensemble de test est le produit cartésien entre les paires de phrases dans les deux langues. Par exemple, avec un ratio de bruit de 60%, 400 des 1,000 paires de phrases sont parallèles, de sorte que seulement 0.04% des 1M de paires de phrases générées par le produit cartésien sont réellement parallèles.

Nos ensembles de données sont normalisés et tokenisés en utilisant les scripts de Moses (Koehn et al., 2007) [2]. La longueur maximale de chaque phrase est fixée à 80 mots. Chaque mot inconnu, c'est-à-dire qui ne se trouve pas dans le vocabulaire, est remplacé par un symbole spécial égal à UNK.

4.1.2. Critères d'évaluation

Pour l'évaluation de la performance de nos modèles, une paire de phrases prédite comme parallèle est correcte si elle est présente dans le sous-ensemble des paires de phrases parallèles (exemples positifs) de l'ensemble de test à évaluer. La précision est la proportion entre $|R|$

2. https ://github.com/moses-smt/mosesdecoder

le nombre de paires de phrases extraites qui sont réellement parallèles dans l'ensemble de test et $|E|$ le nombre de paires de phrases extraites. Le rappel est la proportion entre $|R|$ le nombre de paires de phrases extraites qui sont réellement parallèles dans l'ensemble de test et $|T|$ le nombre de paires de phrases parallèles dans l'ensemble de test. Le score F_1 est la moyenne harmonique de la précision et du rappel :

$$Précision = \frac{|R|}{|E|} \cdot 100, \tag{4.1.1}$$

$$Rappel = \frac{|R|}{|T|} \cdot 100, \tag{4.1.2}$$

$$F_1 = \frac{2 \cdot Précision \cdot Rappel}{Précision + Rappel}. \tag{4.1.3}$$

4.1.3. Système d'extraction de référence (*baseline*)

Comme système d'extraction de référence, nous utilisons un système d'extraction de phrases parallèles développé au RALI [3] par (Bérard, 2014) basé sur les travaux de (Munteanu and Marcu, 2005) et (Smith et al., 2010). Ces systèmes sont encore considérés comme l'état de l'art. Un diagramme du système de référence est présenté à la Figure 4.1. Le système est composé d'un processus de filtrage et de trois modèles ; deux modèles d'alignement de mots et un classificateur d'entropie maximale. Les modèles d'alignement de mots sont entraînés dans les deux directions en utilisant les 500k paires de phrases parallèles de notre ensemble d'entraînement. Pour entraîner le classificateur, comme exemples positifs nous sélectionnons aléatoirement 100k autres paires de phrases parallèles du corpus Europarl qui ne font pas partie des 500k paires de phrases déjà sélectionnées. Pour les exemples négatifs, nous choisissons 100k paires de phrases non parallèles qui ont passé avec succès le processus de filtrage. Une fois le système entraîné, nous l'alimentons de paires d'articles pour obtenir un corpus de paires de phrases potentiellement parallèles.

Nous précisons que (Munteanu and Marcu, 2005) utilise 5k paires de phrases parallèles comme exemples positifs pour entraîner leur classificateur. Durant nos expériences, nous avons obtenus un léger gain en performance en utilisant 100k paires de phrases parallèles au lieu de 5k, mais nous n'avons pas observé de gain significatif en utilisant plus de 100k paires de phrases parallèles. Pour ce type de classificateur, étant donné que nous devons calculer l'ensemble des *features* pour chaque phrase, cette approche devient rapidement coûteuse et longue à utiliser plus le nombre de paires de phrases augmente. Il est important de noter que 500k paires de phrases sont aussi utilisées pour entraîner les modèles d'alignement de mots qui sont au coeur du système pour induire les lexiques bilingues. Ces lexiques bilingues servent

3. http ://rali.iro.umontreal.ca/

comme *features* au classificateur et au processus de filtrage (voir les sections 4.1.3.1, 4.1.3.2 et 4.1.3.3).

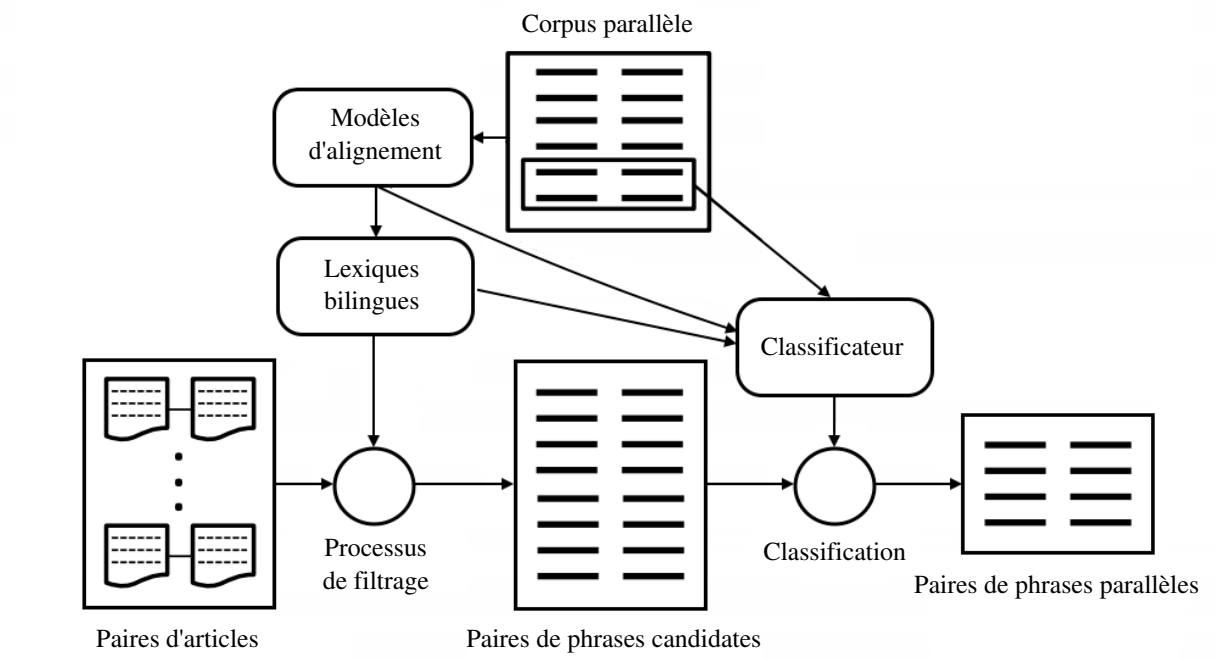


FIGURE 4.1. Système d'extraction de référence.

4.1.3.1. *Processus de filtrage*

Pendant l'apprentissage, un processus de filtrage des paires de phrases est utilisé pour sélectionner un nombre fixe de paires de phrases négatives afin d'entraîner le classificateur d'entropie maximale. L'objectif est de sélectionner des paires de phrases non parallèles ayant un degré de similarité assez élevé pour rendre le classificateur plus robuste au bruit. Le filtrage est également utilisé au moment de faire des prédictions pour filtrer les paires de phrases improbables d'une paire d'articles. Avant le filtrage, l'ensemble des paires de phrases candidates est le produit cartésien entre les phrases de deux articles. Par conséquent, la quantité de paires de phrases à évaluer peut rapidement devenir très élevée. Étant donné que la grande majorité de ces paires de phrases ne sont pas parallèles, le filtrage permet de considérablement réduire le nombre de paires de phrases à évaluer. Au cours de nos expériences, nous avons observé que le processus de filtrage élimine plus de 99% des paires de phrases du produit cartésien de l'ensemble de test, sans lequel le classificateur ne réussirait pas à identifier les phrases parallèles à lui seul.

Le processus de filtrage se fait en deux étapes. Premièrement, il vérifie que le rapport entre la longueur des deux phrases n'est pas supérieur à 2. La deuxième étape consiste à appliquer un filtre de chevauchement de mots (*word overlap*, en anglais) en utilisant les lexiques bilingues induits par les modèles d'alignement. Cette étape s'assure qu'au moins 50% des mots d'une phrase a une traduction directe dans la phrase dans l'autre langue. Chaque paire de phrases qui ne remplit pas ces deux conditions est retirée de l'ensemble d'évaluation.

Il est important de noter que ce processus de filtrage est uniquement appliqué au système de référence. Bien que nous pourrions utiliser ce processus de filtrage afin de sélectionner des exemples négatifs pour faire l'entraînement de notre approche, nous préférons ne pas avoir besoin d'utiliser des modèles d'alignement de mots.

4.1.3.2. *Modèles d'alignement*

Les tables de traduction et d'alignement sont estimées en utilisant le modèle d'alignement HMM de (Vogel et al., 1996). Ces tables de probabilités sont nécessaires pour mesurer la valeur des nombreuses fonctions d'alignement utilisées comme *features* dans le classificateur. Un modèle IBM 2 utilise ensuite ces tables de probabilités pour effectuer l'alignement des mots en alignant chaque mot de la phrase source à la position d'un mot de la langue cible en maximisant le produit des probabilités de traduction et d'alignement. Étant donné w_i^s un mot de la phrase de la langue source de N mots, il est aligné à la position a_i avec w_j^c un mot de la phrase de la langue cible de longueur M selon :

$$a_i = \underset{j \in \{0,\ldots,M\}}{\operatorname{argmax}} \, t(w_i^s | w_j^c) \cdot q(j|i,N,M), \tag{4.1.4}$$

où $t(w_i^s|w_j^c)$ est la probabilité de traduction, $q(j|i,N,M)$ est la probabilité d'alignement et $a_i = 0$ pour un mot qui n'a pas d'alignement.

Les tables de traduction sont aussi utilisées pour induire les lexiques bilingues servant comme filtre de chevauchement de mots à la sélection de paires de phrases candidates. Pour réduire le bruit dans les nombreuses paires de mots qui ont été alignées, nous gardons les paires de mots des tables de traduction estimées ayant une probabilité supérieure à 10%. Pour entraîner nos modèles d'alignement de mots, nous utilisons le *toolkit* GIZA++ (Och and Ney, 2003) [4].

4.1.3.3. *Classificateur d'entropie maximale*

Le classificateur utilise 31 *features* qui sont basées sur les travaux de (Munteanu and Marcu, 2005) and (Smith et al., 2010). Il s'appuie sur des *features* d'alignement des mots entre une

4. http ://www.statmt.org/moses/giza/GIZA++.html

paire de phrases dans deux langues, tels que le nombre et le pourcentage des mots qui sont connectés, le nombre et le pourcentage des mots qui ne sont pas connectés, les trois plus grandes fertilités, le pourcentage des mots source ayant une fértilité de 1, 2, 3 ou plus, la longueur du plus long segment de mots connectés, la longueur du plus long segment de mots qui ne sont pas connectés, le logarithme de la probabilité d'alignement, ainsi que des *features* généraux, tels que la longueur des phrases, la différence des longueurs, le ratio des longueurs et le pourcentage des mots de qui ont une traduction de l'autre côté. La classificateur donne la probabilité qu'une paire de phrases soit parallèle ou non. Une paire de phrases est classée comme parallèle si le classificateur donne une probabilité supérieure ou égale à un seuil de décision ρ qui doit être fixé.

4.1.4. Détails sur l'entraînement des modèles

Les modèles de notre approche sont implémentés avec TensorFlow (Abadi et al., 2016). Nous utilisons un BiRNN siamois avec une seule couche cachée dans chaque direction (voir section 3.2). La dimension des *word embeddings* est égale à 512 et les états récurrents contiennent 512 unités cachées. Nous utilisons un GRU comme fonction d'activation récurrente, car il a constamment surperformé un LSTM par une petite marge durant nos expériences. La couche cachée du réseau de neurones *feed-forward* contient 256 unités cachées, pour enfin émettre un score de probabilité comme prédiction finale. Nous initialisons tous les paramètres uniformément en utilisant la fonction d'initialisation par défaut de TensorFlow, sauf pour tous les biais initialisés à zéro. Pour entraîner nos modèles, nous utilisons l'optimiseur Adam avec un taux d'apprentissage de 0.0002 et des mini-batches de 128 exemples pour un total de 15 époques d'entraînement. Pour éviter le problème d'explosion du gradient, nous normalisons la norme du gradient telle qu'elle ne soit pas plus grande que 5. Nous appliquons du *dropout* pour éviter le surapprentissage avec une probabilité de 0.2 et 0.3 sur les connexions d'entrées et de sorties non récurrentes (Zaremba et al., 2014). L'entraînement est effectué sur un seul *GPU*.

4.1.5. Nombre d'exemples négatifs

Pour que notre système soit en mesure d'apprendre à différencier les paires de phrases en relation de traduction à celles qui ne le sont pas, nous devons générer et ajouter des exemples négatifs dans notre ensemble d'entraînement (voir section 3.1). Dans cette section, nous évaluons la performance de notre approche en fonction de m le nombre d'exemples négatifs générés par paire de phrases. Nous entraînons 10 modèles avec $m \in \{1, \ldots, 10\}$, de sorte qu'avec $m = 1$ et $m = 10$ un modèle est respectivement entraîné sur 1M et 5.5M de paires de phrases par époque avec un ratio du nombre d'exemples positifs et négatifs de 50% et 0.09%. Nous rappelons qu'au début de chaque nouvelle époque nous échantillonnons des nouveaux

exemples négatifs, permettant au système d'apprendre sur un plus large ensemble d'exemples. D'un autre côté, plus la valeur de m augmente, plus l'ensemble d'entraînement devient déséquilibré. Nous savons qu'en pratique la tâche d'extraction de phrases parallèles est un problème de classification hautement déséquilibré avec les paires de phrases non parallèles représentant la classe majoritaire. Malgré qu'un ensemble d'entraînement déséquilibré n'est pas souhaité puisqu'un classificateur entraîné sur un tel ensemble aura généralement tendance à prédire la classe majoritaire et d'avoir une mauvaise précision, l'impact du niveau de déséquilibre dans l'ensemble d'entraînement sur la performance du système n'est toujours pas clair. A priori, nous préférons un ensemble d'entraînement qui est le moins déséquilibré possible.

Nous évaluons la relation entre le déséquilibre et la performance de notre approche sur nos ensembles de test newstest2012 avec des ratios de bruit de 0%, 50%, 90%. La Figure 4.2 montre la valeur du score F_1 pour les paires de phrases extraites de ces 3 ensembles de test par nos 10 modèles. Chaque score F_1 présenté est celui où la valeur du seuil de décision ρ est fixée de manière à le maximiser, c'est-à-dire la valeur qui maximise l'aire sous la courbe de rappel et de précision. En principe, nous nous attendons à ce que la performance d'un système se détériore pour des valeurs de m élevées (pour un ensemble d'entraînement fortement déséquilibré). En effet, pour des valeurs supérieures à $m = 7$ nous remarquons cette tendance avec les ensembles bruités qui ont un ratio de 0% et 50%, alors que le meilleur modèle pour l'ensemble fortement bruité avec un ratio de 90% est celui avec $m = 6$. Pour notre approche, nous voyons qu'avoir un ensemble d'entraînement parfaitement équilibré avec $m = 1$ n'est pas sa solution optimale. Au contraire, avoir un ensemble d'entraînement

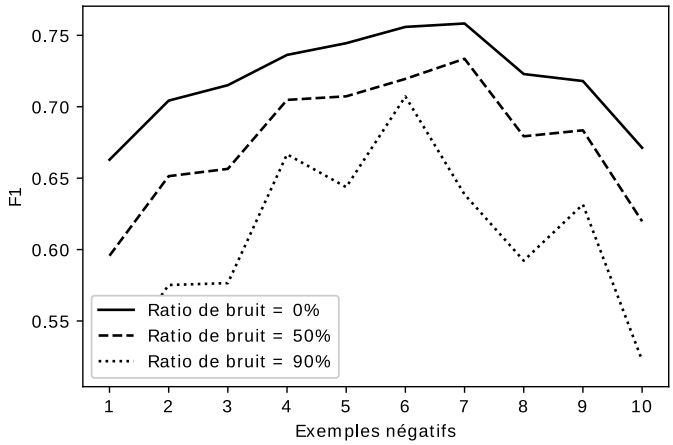


FIGURE 4.2. Score F_1 de notre approche en fonction du nombre d'exemples négatifs générés par paire de phrases parallèles. Les modèles sont évalués sur newstest2012 avec des ratios de bruit de 0%, 50% et 90%.

déséquilibré peut améliorer considérablement la performance.

À partir de ces observations, dans les expériences suivantes nous entraînons des modèles avec une valeur de m fixée à 6.

4.1.6. Comparaison sur corpus parallèles bruités

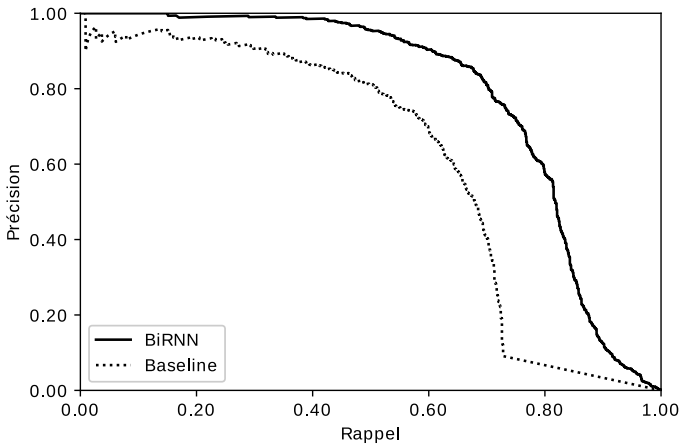
Dans cette section, nous mesurons la précision, le rappel et le score F_1 afin de comparer notre approche avec le système d'extraction de référence. BiRNN est notre approche entraînée avec 6 exemples négatifs par paire de phrases parallèles, alors que Baseline représente le système de référence (voir section 4.1.3).

Le Tableau 4. I présente les valeurs des critères d'évaluation pour les deux systèmes évalués sur les ensemble de test de newstest2012 et d'Europarl avec des ratios de bruit de 0%, 50% et 90%. Nous voyons que BiRNN est en mesure de constamment dépasser de manière significative les résultats obtenus avec le système de référence. En utilisant newstest2012 comme ensembles de test hors domaine, notre approche obtient une amélioration du score F_1 par rapport au système de référence de 10.99%, 13.03% et 23.53% pour les ensembles de test avec des ratios de bruit de 0%, 50% et 90%, respectivement. Les courbes de rappel et de précision des deux systèmes évalués sur ces trois ensembles de test sont présentées dans la Figure 4.3. Celles-ci illustrent la consistance de notre approche, alors que la performance du système de référence se dégrade considérablement lorsque le niveau de bruit devient élevé.

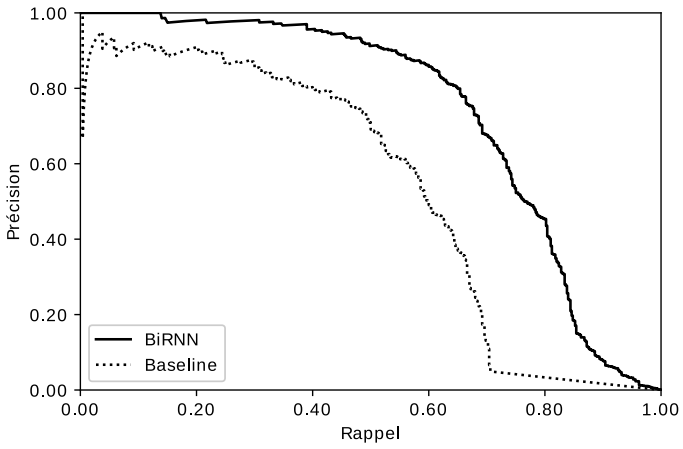
Les résultats obtenus sur les ensembles de test d'Europarl sont marquants. Notre approche réussit à obtenir des scores F_1 au delà de 95% sur les trois ensembles de test venant du même domaine. D'un autre côté, les scores obtenus par le système de référence sur l'ensemble de

		newstest2012				**Europarl**			
Bruit	**Modèle**	**P (%)**	**R (%)**	**F_1 (%)**	ρ	**P (%)**	**R (%)**	**F_1 (%)**	ρ
0%	BiRNN	**83.72**	**68.90**	**75.79**	0.99	**99.26**	**93.50**	**96.29**	0.99
	Baseline	73.88	57.70	64.80	0.93	87.72	84.30	85.98	0.98
50%	BiRNN	**79.95**	**65.40**	**71.95**	0.99	**98.32**	**93.60**	**95.90**	0.99
	Baseline	73.43	49.20	58.92	0.98	80.19	82.60	81.38	0.99
90%	BiRNN	**79.01**	**64.00**	**70.72**	0.99	**97.94**	**95.00**	**96.45**	0.99
	Baseline	53.85	42.00	47.19	0.99	63.89	69.00	66.35	0.99

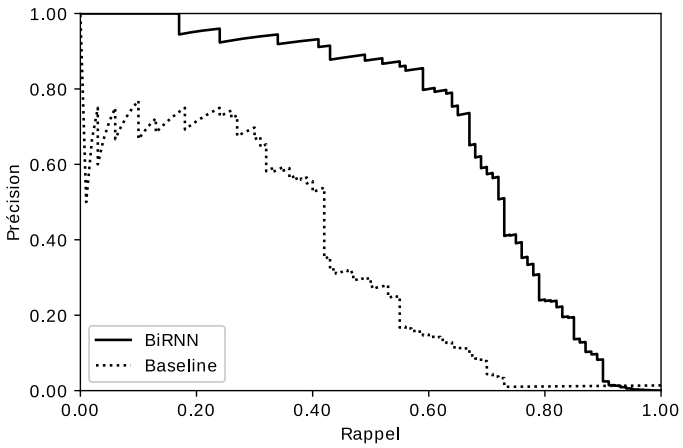
TABLEAU 4. I. Précision (P), rappel (R) et score F_1 où le seuil de décision ρ maximise l'aire sous la courbe de rappel et de précision sur les ensembles de test avec des ratios de bruit de 0%, 50% et 90%.



(A) Ratio de bruit de 0%.



(B) Ratio de bruit de 50%.

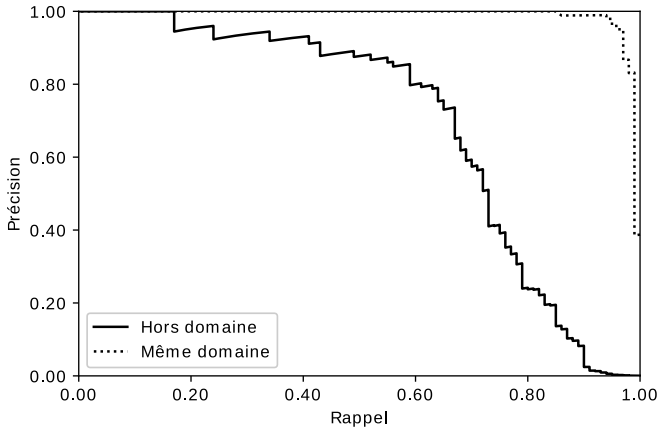


(C) Ratio de bruit de 90%.

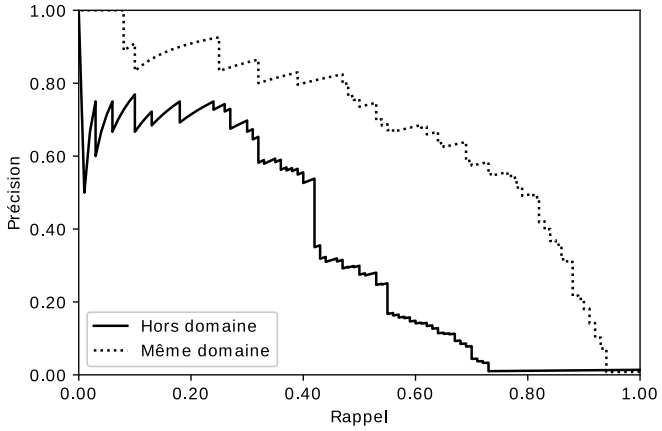
FIGURE 4.3. Courbes de rappel et de précision des systèmes évalués sur newstest2012.

test sans bruit sont dans le même ordre de grandeur que ceux retrouvés dans les travaux évaluant ce système. Cependant, en augmentant le nombre de phrases non parallèles dans l'ensemble de test nous observons encore une importante détérioration de sa performance.

Dans la Figure 4.4, nous comparons respectivement les courbes de rappel et de précision des sytèmes évalués sur les ensembles de test de newstest2012 et d'Europarl avec un ratio de bruit de 90%. Lorsque nous avons des textes non parallèles traitant des sujets différents que ceux retrouvés dans l'ensemble d'entraînement, nous remarquons que BiRNN s'adapte largement mieux que le système de référence pour extraire les phrases parallèles. Sachant que les modèles d'alignement de mots ainsi que le classificateur du système de référence sont



(A) BiRNN.



(B) Baseline.

FIGURE 4.4. Courbes de rappel et de précision des systèmes évalués sur les ensembles de test hors domaine (newstest2012) et du même domaine (Europarl) avec un ratio de bruit de 90%.

entraînés sur les données d'Europarl, en l'évaluant sur un ensemble de test bruité venant du même corpus nous nous attendions à ce qu'il performe mieux que les résultats obtenus.

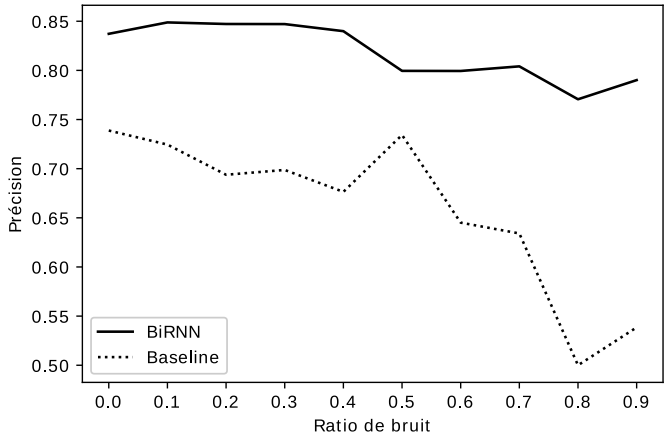
Nous précisons à nouveau que notre approche évalue les 1M paires de phrases du produit cartésien des phrases de chaque ensemble de test, alors que le système de référence applique un processus de classification à deux étapes en filtrant d'abord les paires de phrases du produit cartésien (voir section 4.1.3.1) pour réduire le nombre de paires de phrases à évaluer avec le classificateur. Pour avoir une idée de l'impact du filtrage sur le nombre de paires de phrases à évaluer ainsi que sur la performance du système de référence, nous l'évaluons sans processus de filtrage sur des ensembles de test de newstest2012. Les résultats sont présentés dans le Tableau 4. II. En effet, nous voyons que le processus de filtrage élimine la grande majorité du total des paires de phrases à évaluer, en retirant jusqu'à 99.28% des paires. Ces résultats donnent un bon aperçu sur l'importance du processus de filtrage pour le bon fonctionnement du système de référence. Nous voyons que le processus de filtrage est un élément clé et intégré du système de référence et il serait injuste de directement comparer ces performances avec celles obtenues par la nôtre. Toutefois, comme travail futur il pourrait être intéressant d'ajouter le processus de filtrage à notre approche [5]. Bien que le processus de filtrage retire plus de 99% des paires de phrases à évaluer, la comparaison des résultats du système de référence entre les Tableaux 4. I et 4. II réflète l'efficacité du processus de filtrage à écarter les paires de phrases qui ne sont pas parallèles.

	Paires			**newstest2012**		
Bruit	**Avec filtre**	**Sans filtre**	**Δ (%)**	**P (%)**	**R (%)**	**F_1 (%)**
0%	8,053	1,000,000	99.19	28.69	17.50	21.74
50%	7,248	1,000,000	99.27	28.38	13.00	17.83
90%	7,215	1,000,000	99.28	10.00	13.00	11.30

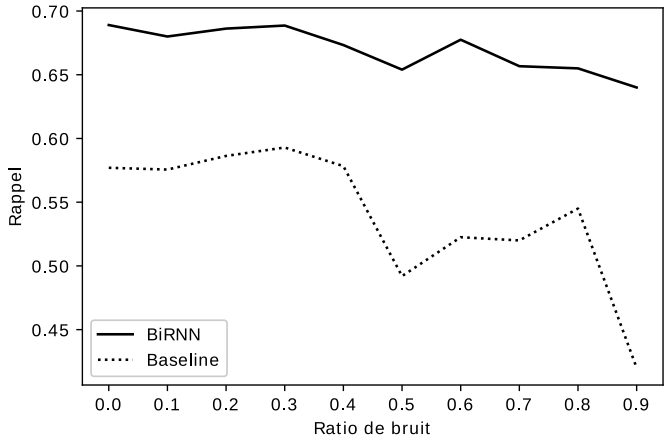
TABLEAU 4. II. Performance du système de référence sans le processus de filtrage. Δ est le pourcentage des paires de phrases retirées du produit cartésien.

Afin de mieux évaluer la robustesse de notre approche sur différents degrés de comparabilité de corpus comparables, dans la Figure 4.5 nous comparons la précision, le rappel et le score F_1 à mesure que le ratio de bruit augmente dans notre ensemble de test newstest2012. Une tendance s'observe pour les deux systèmes : il devient de plus en plus difficile d'identifier des phrases parallèles lorsque le nombre de phrases non parallèles augmente.

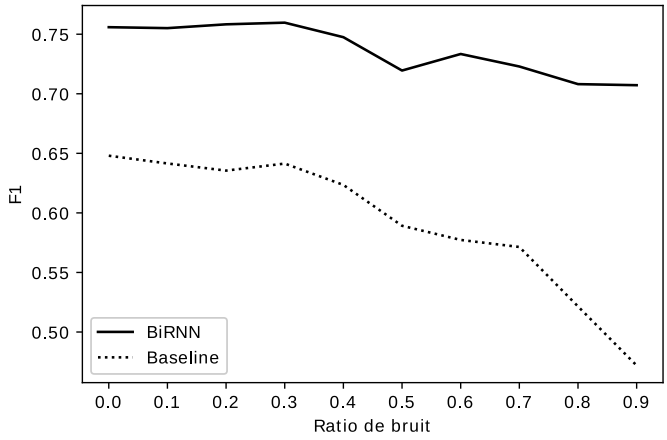
5. Comme nous avons déjà mentionné, nous préférons ne pas avoir à utiliser des ressources ou des modèles externes.



(A) Précision.



(B) Rappel.



(C) Score F_1.

FIGURE 4.5. Performance des systèmes en fonction du ratio de bruit dans newstest2012.

Cependant, nous voyons que notre approche à base de réseaux de neurones performe mieux que le système de référence sur toute la ligne. Contrairement au système de référence, la performance générale de notre méthode reste relativement stable, particulièrement pour la précision. Lorsque le ratio de bruit est supérieur à 50%, la précision du système de référence chute de manière considérable. De plus, la présence élevée de phrases non parallèles impacte beaucoup moins le rappel de notre approche que celui du système de référence. À un tel niveau de bruit, nous croyons que l'ensemble de test est plus représentatif des textes que nous pouvons trouver dans des corpus comparables.

Tous les critères d'évaluation sont encore présentés pour les paires de phrases extraites lorsque la valeur du seuil de décision ρ est fixée à celle qui maximise l'aire sous la courbe de rappel et de précision. Ces niveaux de performance sont donc des bornes supérieures. En pratique, nous devons fixer cette valeur. Il est alors préférable d'avoir un système ou la valeur optimale est stable. Pendant nos expériences, nous avons remarqué que le seuil de décision optimal de BiRNN se trouvait toujours alentour de $\rho = 0.99$, alors que celui du système de référence variait considérablement d'un ensemble de test à un autre. Certains peuvent croire que la précision des paires extraites est plus importante que le rappel et qu'un système donnant le meilleur score F_1 n'est pas la solution optimale. À cet égard, (Goutte et al., 2012) constate que les systèmes SMT sont robustes au bruit dans l'alignement des données d'entraînement et que le rappel peut être encore plus important que la précision. Cependant, il est possible que cela ne s'applique pas pour les systèmes NMT. Ces systèmes sont fortement basés sur des représentations de distributions sémantiques, où la précision pourrait être le score à prioriser. Dans tous les cas, par comparaison avec le système de référence, nous voyons que notre approche donne toujours une meilleure précision pour une valeur de rappel plus grande. Cela signifie qu'établir la valeur du seuil de décision dans le but d'obtenir approximativement une précision désirée conduira à un plus grand nombre de phrases parallèles de haute qualité que le système de référence. En effet, la valeur du seuil de décision a directement un impact sur la qualité et la quantité de paires de phrases extraites. Dans notre cas, étant donné que nous sommes en présence d'ensembles de données avec des classes fortement déséquilibrées, nous recommandons d'utiliser une valeur très élevée $\rho \approx 0.99$ afin de réduire le nombre de faux positifs.

4.2. Évaluation sur systèmes de traduction automatique

4.2.1. Données

Les données utilisées dans cette expérience sont aussi des ensembles anglais et français de WMT'15. L'ensemble d'entraînement utilisé pour entraîner les systèmes SMT et NMT de références est notre ensemble contenant 500k paires de phrases parallèles du corpus

Europarl décrit à la section 4.1.1. Nous avons aussi entraîné des systèmes de traduction sur le total des 2M paires de phrases contenues dans Europarl afin de comparer la valeur ajoutée d'utiliser des paires de phrases extraites par rapport à un large corpus parallèle. Le corpus newstest2012 est notre ensemble de validation et nous évaluons nos modèles sur le corpus newstest2013, qui contient 3,000 paires de phrases qui ne font pas partie de l'ensemble d'entraînement. Les corpus comparables que nous utilisons pour extraire des phrases parallèles sont 919k paires d'articles anglais-français obtenues des dumps Wikipédia [6] organisées par (Bérard, 2014) que nous gardons à l'interne au RALI.

Nos ensembles de données sont normalisés et tokenisés en utilisant les scripts de Moses. La longueur maximale de chaque phrase est fixée à 60 mots. Pour les systèmes de traduction neuronale, la taille du vocabulaire pour les deux langues est limitée aux 150k mots les plus fréquents de l'ensemble d'entraînement utilisé. Tous les mots inconnus sont remplacés par le symbole UNK.

4.2.2. Critères d'évaluation

Pour évaluer la performance de traduction des différents systèmes, nous utilisons comme métrique le score BLEU (Papineni et al., 2002) en utilisant le script multi-bleu de Moses. Ce critère d'évaluation est compris entre 0 et 1 et mesure la moyenne géométrique des scores de précision modifiée des n-grammes. La précision modifiée des n-grammes d'une traduction se définit comme :

$$P_n = \frac{\sum_{S \in C} \sum_{n\text{-gramme} \in S} \hat{c}(n\text{-gramme})}{\sum_{S \in C} \sum_{n\text{-gramme} \in S} c(n\text{-gramme})}, \tag{4.2.1}$$

où C est l'ensemble de toutes les phrases traduites, S est l'ensemble des n-grammes qui se trouvent dans une phrase de C, $c(n\text{-gramme})$ est le nombre d'occurences du n-gramme et $\hat{c}(n\text{-gramme})$ est :

$$\hat{c}(n\text{-gramme}) = \min(c(n\text{-gramme}), c_{réf}(n\text{-gramme})). \tag{4.2.2}$$

$c_{réf}(n\text{-gramme})$ est le nombre d'occurences du n-gramme dans la phrase de référence. En d'autres mots, il s'agit de comparer les n-grammes d'une traduction qui se retrouvent dans ceux de la phrase de référence. Les scores de précision modifiée des n-grammes sont typiquement calculés pour pour les n-grammes d'ordres 1 à 4. Le score BLEU est :

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{4} w_n \log P_n\right), \tag{4.2.3}$$

6. https ://dumps.wikimedia.org/

où $w_n = 1/4$ et BP est une fonction qui pénalise les phrases trop courtes par rapport à la référence :

$$BP = \min\left(1, \frac{|traduction|}{|référence|}\right). \tag{4.2.4}$$

4.2.3. Systèmes de traduction automatique

4.2.3.1. *Traduction automatique statistique*

L'objectif d'un système SMT est de générer une phrase cible $\mathbf{s}^c$ qui est la traduction d'une phrase source $\mathbf{s}^s$. Parmi toutes les phrases possibles dans la langue cible, le système choisit celle avec la probabilité conditionnelle la plus élevée, tel que :

$$\begin{aligned} \hat{\mathbf{s}}^c &= \underset{\mathbf{s}^c}{\operatorname{argmax}}\, p(\mathbf{s}^c|\mathbf{s}^s), \\ &= \underset{\mathbf{s}^c}{\operatorname{argmax}}\, p(\mathbf{s}^s|\mathbf{s}^c) \cdot p(\mathbf{s}^c), \end{aligned} \tag{4.2.5}$$

où $p(\mathbf{s}^s|\mathbf{s}^c)$ est un modèle de traduction de mots et $p(\mathbf{s}^c)$ est un modèle de langue dans la langue cible. La première génération de système de traduction statistique à base de mots a été proposée par (Brown et al., 1993) avec les modèles IBM, où un lexique bilingue est utilisé pour entraîner le modèle de traduction et un corpus monolingue dans la langue cible pour entraîner le modèle de langue.

De nos jours, lorsque nous utilisons un système SMT, il est plus commun d'utiliser un système à base de segments (Koehn et al., 2003) (*phrase-based*, en anglais) entraîné sur un corpus parallèle. Un segment est défini comme une séquence de mots dans la langue source qui doivent être traduits en une séquence de mots dans la langue cible. Une première étape consiste à utiliser des modèles d'alignement pour extraire les paires de segments cohérents d'un corpus parallèle. Un probabilité de traduction estimée par la fréquence relative doit être assignée à chaque paire de segment $(\bar{\mathbf{s}}_i^s, \bar{\mathbf{s}}_i^c)$, telle que :

$$p(\bar{\mathbf{s}}_i^s|\bar{\mathbf{s}}_i^c) = \frac{c(\bar{\mathbf{s}}_i^s, \bar{\mathbf{s}}_i^c)}{\sum_k c(\bar{\mathbf{s}}_i^s, \bar{\mathbf{s}}_k^c)}, \tag{4.2.6}$$

où $c(\cdot)$ est une fonction de comptage. Le nombre de segments possibles peut typiquement avoir une valeur très élevée, alors il est commun de limiter la longueur des segments à 7 mots. Un modèle de réordonnancement $d(a_i - b_{i-1} - 1)$, où a_i est la position du premier mot du segment source aligné au i-ième segment traduit et b_{i-1} est la position du dernier mot du segment source aligné au (i-1)ème segment traduit, mesure une probabilité de distorsion relative pour réorganiser l'ordre des segments d'une phrase traduite[7] . Comme pour un système de traduction statistique à base de mots, un modèle de langue n-gramme dans la

7. En pratique, les systèmes à base de segments obtiennent des meilleurs résultats avec un modèle de réordonnancement lexical.

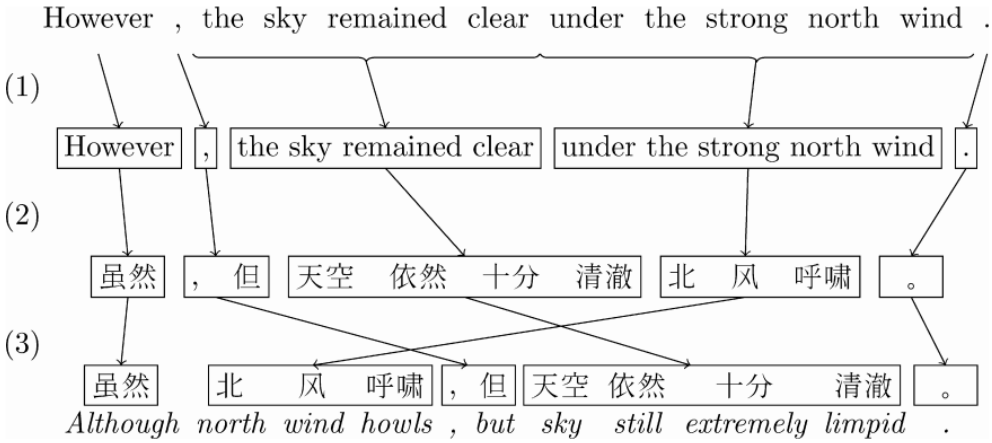


FIGURE 4.6. Exemple des étapes de traduction pour un système de traduction statistique à base de segments (Lopez (2008)).

langue cible $p(\mathbf{s}^c)$ est entraîné. La meilleure phrase traduite est obtenue selon le modèle suivant :

$$\begin{aligned}\hat{\mathbf{s}}^c = \underset{\mathbf{s}^c}{\operatorname{argmax}} \Bigg\{ \exp\Bigg(& \lambda_1 \sum_{i=1}^{S} \log p(\bar{\mathbf{s}}_i^s | \bar{\mathbf{s}}_i^c) + \\ & \lambda_2 \sum_{i=1}^{S} \log d(a_i - b_{i-1} - 1) + \\ & \lambda_3 \sum_{i=1}^{N} \log p(\mathbf{s}_i^c | \mathbf{s}_{i-n+1}^c, \dots \mathbf{s}_{i-1}) \Bigg) \Bigg\}\end{aligned} \tag{4.2.7}$$

où S et N sont respectivement le nombre de segments et le nombre de mots dans la phrase traduite (ou cible). Les paramètres λ_i sont généralement optimisés sur un ensemble de validation. Plusieurs autres fonctions et *features* peuvent être ajoutés au système, particulièrement des alignements bidirectionnels en ajoutant un modèle de traduction par segments $p(\bar{\mathbf{s}}^c | \bar{\mathbf{s}}^s)$, un modèle de langue dans la langue source $p(\mathbf{s}^s)$ ou un modèle de pénalité par rapport à la longueur des segments. La Figure 4.6 est une illustration adaptée de (Lopez, 2008) montrant le processus de traduction d'un système à base de segments. Ce processus se fait en trois étapes :

1. Une phrase source est segmentée en séquences de mots (segments).
2. Chaque segment est traduit.
3. L'ordre des segments traduits est réorganisé pour obtenir une phrase traduite dans la langue cible.

4.2.3.2. *Traduction automatique neuronale*

L'objectif d'un système NMT est identique à celui d'un système SMT, soit de trouver la phrase cible qui maximise la probabilité conditionnelle $p(\mathbf{s}^c | \mathbf{s}^s)$. Cependant, au lieu

d'utiliser une suite de modèles comme avec les systèmes SMT, un modèle de traduction est appris par des réseaux de neurones en maximisant la probabilité conditionnelle des paires de phrases d'un corpus parallèle. Un système NMT consiste généralement en une architecture encodeur-décodeur, où la première composante encode une phrase source en une représentation vectorielle et la seconde la décode en une phrase traduite dans la langue cible.

Les approches de (Sutskever et al., 2014; Cho et al., 2014) proposent de traiter séquentiellement les mots w_t^s d'une phrase source avec un RNN pour encoder la phrase en une représentation vectorielle, tel que :

$$\mathbf{h_t} = f_{enc}(w_t^s, \mathbf{h}_{t-1}), \tag{4.2.8}$$

où $f_{enc}(\cdot)$ peut être n'importe quelle fonction d'activation récurrente (voir section 2.2). Après avoir traité une phrase source de longueur N jusqu'à la fin, le dernier état récurrent $\mathbf{h}_N$ résume l'information de cette phrase au complet en une seule représentation vectorielle, que nous appelons le vecteur de contexte $\mathbf{c}$. Le décodeur, lui aussi sous forme d'un RNN, est entraîné simultanément en utilisant la phrase cible et agit comme un modèle de langue neuronal conditionné sur le vecteur de contexte et les mots précédents pour générer une phrase traduite. La représentation vectorielle de l'état récurrent du décodeur au pas de temps t est donc égale à :

$$\mathbf{z_t} = f_{dec}(w_{t-1}^c, \mathbf{z}_{t-1}, \mathbf{c}), \tag{4.2.9}$$

où $f_{dec}(\cdot)$ est une fonction d'activation récurrente. Une fois que l'état récurrent du décodeur est mis à jour, nous pouvons calculer la probabilité conditionnelle pour chaque mot qui se trouve dans le vocabulaire de la langue cible :

$$p(w_t^c | w_1^c, \ldots, w_{t-1}^c, \mathbf{c}) = g(w_{t-1}^c, \mathbf{z}_t, \mathbf{c}), \tag{4.2.10}$$

où $g(\cdot)$ est une fonction paramétrisée par un réseau de neurones *feed-forward* qui retourne un score de probabilité.

Au lieu d'utiliser le même vecteur de contexte $\mathbf{c}$ égal à $\mathbf{h}_N$ pour chaque pas de temps lors du décodage, (Bahdanau et al., 2014) propose d'utiliser un mécanisme d'attention permettant de calculer un vecteur de contexte différent à chaque pas de temps $\mathbf{c}_t$ qui dépend de toute la séquence $\{\mathbf{h}_1, \ldots, \mathbf{h}_N\}$. En plus de ne plus avoir besoin de représenter chaque phrase source en un seul vecteur de contexte, cela permet au système de se concentrer sur les segments d'information pertinents de la phrase source pour la génération du prochain mot.

4.2.4. Détails sur l'entraînement des modèles

Nos systèmes SMT sont des systèmes à base de segments que nous avons entraînés avec le *toolkit* Moses. Premièrement, nous utilisons GIZA++ pour entraîner les modèles d'alignement de mots dans les deux directions. Ensuite, les segments et les réordonnancements lexicaux sont extraits en utilisant les valeurs par défaut de Moses. Les modèles de langues sont des modèles 5-grammes appris en utilisant le *toolkit* KenLM (Heafield, 2011) [8] sur les parties monolingues du même corpus parallèle fourni pour l'apprentissage des modèles de traduction. Les paramètres λ_i sont optimisés sur les 3,003 paires de phrases parallèles du corpus newstest2012.

Pour entraîner nos systèmes NMT, nous utilisons l'implémentation PyTorch de OpenNMT (Klein et al., 2017) [9]. Tous nos systèmes consistent en un BiRNN avec une couche cachée dans chaque direction. Nous utilisons un LSTM comme fonction d'activation récurrente. Le type de mécanisme d'attention est un réseau de neurones *feed-forward*. Les dimensions des *word embeddings* et des états récurrents pour l'encodeur et le décodeur sont toutes égales à 256. Les systèmes sont entraînés sur 10 époques [10] en utilisant SGD avec un taux d'apprentissage initial égal à 1.0 qui est linéairement décroissant après chaque époque. Une mini-batch contient 64 paires de phrases. La norme du gradient est tronquée de sorte qu'elle ne soit pas supérieure à 5. Durant l'apprentissage, une probabilité *dropout* égale à 0.3 est appliquée sur les connexions de sorties non récurrentes.

Pour chacune des approches SMT et NMT, nous avons entraîné 14 systèmes de traduction automatique. Le premier système [11], servant de système de traduction de référence, est entraîné sur notre ensemble de 500k paires de phrases parallèles du corpus Europarl. Le deuxième est entraîné sur les 2M paires de phrases parallèles d'Europarl. Ce système sert à comparer la valeur ajoutée des paires de phrases extraites de Wikipédia par rapport à l'utilisation d'un large corpus d'entraînement que nous pouvons facilement avoir à la portée de la main. Les autres systèmes utilisent les $\{250\text{k}, 500\text{k}, \ldots, 1.25\text{M}, 1.5\text{M}\}$ meilleures paires de phrases extraites de Wikipédia par notre approche ou le système d'extraction de référence en les ajoutant aux 500k paires de phrases parallèles utilisées pour le système de traduction de référence. Par meilleures paires de phrases extraites, nous voulons dire l'ensemble des paires de phrases extraites triées par ordre décroissant en fonction du score de probabilité estimé par un système d'extraction.

8. https ://github.com/kpu/kenlm
9. https ://github.com/OpenNMT/OpenNMT-py
10. Bien que nous pouvons entraîner nos systèmes NMT sur plus d'époques afin d'obtenir des meilleurs résultats, l'objectif de cette expérience est principalement de comparer la qualité entre les ensembles de paires de phrases extraites par BiRNN et le système d'extraction de référence.
11. C'est-à-dire le premier système pour chaque approche et non pas du total des 28 systèmes de traduction.

Langue	Modèle	Paires	Mots	Longueur
Anglais	BiRNN	1,500,000	33,781,379	22 ± 14
	Baseline	1,500,000	28,586,878	19 ± 11
Français	BiRNN	1,500,000	37,741,039	25 ± 15
	Baseline	1,500,000	30,838,736	20 ± 12

TABLEAU 4. III. Statistiques sur les 1.5M de paires de phrases extraites des paires d'articles anglais-français de Wikipédia. Longueur est la moyenne et l'écart-type du nombre de mots dans les phrases.

4.2.5. Extraction de phrases parallèles et comparaison sur systèmes de traduction automatique

L'objectif de l'extraction de phrases parallèles est de créer des nouveaux corpus parallèles, ce qui nous permet de couvrir plus de domaines afin d'améliorer la généralisation des systèmes de traduction automatique. Pour justifier l'utilité de notre approche, nous extrayons des phrases parallèles des articles de Wikipédia anglais-français et évaluons leur qualité en mesurant le score BLEU sur des systèmes SMT et NMT.

Comme nous avons déjà mentionné, la quantité et la qualité des paires de phrases extraites dépendent de la valeur du seuil de décision. Pour mieux comparer nos approches, nous utilisons des sous-ensembles des 1.5M de paires de phrases extraites les mieux classées par chaque système d'extraction, en ne gardant que les paires de phrases dans lesquelles les deux phrases contiennent au moins 3 mots. Le Tableau 4. III montre les informations sur les nouveaux corpus parallèles que nous avons extraits. Nous remarquons une tendance pour le système de référence à retourner des phrases moins longues que celles de BiRNN. Nous avons calculé le taux de recouvrement entre ces deux ensembles et nous sommes surpris qu'il soit seulement de 10%. À première vue, cela pourrait signifier que les deux approches ne soient pas vraiment complémentaires. Cependant, nous soupçonnons que le taux de recouvrement pourrait augmenter en utilisant un plus grand ensemble que les 1.5M de paires de phrases les mieux classées. Une étude plus approndie serait intéressante à faire comme travail futur.

Le Tableau 4. IV illustre les scores BLEU obtenus sur les 14 systèmes de traduction automatique que nous avons entraînés pour chacune des approches SMT et NMT. Les lignes Europarl représentent des systèmes de références que nous avons entraînés avec ce corpus. Des systèmes de traduction ont été entraînés en ajoutant aux 500k paires de phrases d'Europarl les $\{250\text{k}, 500\text{k}, 750\text{k}, 1\text{M}, 1.25\text{M}, 1.5\text{M}\}$ paires de phrases extraites les mieux classées par notre approche et le système d'extraction de référence. Les nombres entre parenthèses sont les gains en score BLEU obtenus par rapport aux systèmes de références

entraînés sur 500k paires de phrases d'Europarl. Nous rappelons que les systèmes SMT ont été optimisés sur un ensemble de validation. Les systèmes NMT ont été entraînés sur 10 époques. Même si l'évaluation des systèmes NMT sur l'ensemble de validation continuait de s'améliorer et que nous aurions pu continuer l'entraînement afin d'avoir des meilleures performances, cela n'est pas l'objectif de cette expérience.

Nous voyons qu'ajouter les sous-ensembles des corpus parallèles extraits par les deux systèmes donne des gains significatifs par rapport aux systèmes de références entraînés avec Europarl. En ajoutant les 1.5M de paires de phrases extraites par notre approche, les gains en scores BLEU sont respectivement de 3.71 et 9.47 sur les systèmes SMT ou NMT. Pour les systèmes entraînés sur 2M de paires de phrases, avec un nombre de paires de phrases égal, notre approche a des gains de 1.98 et 1.70 par rapport aux systèmes de références SMT et NMT. En ajoutant 500k paires de phrases extraites aux 500k paires de phrases d'Europarl, ces systèmes entraînés avec 1M de paires de phrases performent mieux que les systèmes entraînés avec 2M de paires de phrases d'Europarl. Avec la moitié des données nous obtenons une meilleure performance de traduction. En étendant le vocabulaire d'un ensemble d'entraînement, nous voyons que l'extraction de phrases parallèles peut permettre aux systèmes de traduction de mieux généraliser et de mieux traiter les mots inconnus. Toutefois, il est connu que la taille du vocabulaire utilisable des systèmes de traduction automatique reste limitée en raison de la capacité en mémoire, du temps de calcul et aussi pour généraliser hors vocabulaire. Pour répondre à ce problème, au lieu d'utiliser les mots comme unité lexicale il est recommandé pour les systèmes NMT d'utiliser les caractères (Chung et al., 2016) où des segments de mots (Sennrich et al., 2015).

Nous nous intéressons aussi à comparer les gains en performance de traduction entre les paires de phrases extraites par notre approche et celle du système d'extraction de référence. Malgré qu'insérer des phrases non parallèles dans 1,000 paires de phrases du corpus newstest2012 n'est pas représentatif des paires de phrases retrouvées dans les articles de Wikipédia, avec les résultats que nous avons obtenus à la section 4.1.6 nous nous attendions à voir un gain plus significatif entre notre approche et le système d'extraction de référence. Toutefois, en plus d'être plus flexible que le système d'extraction de référence, notre approche réussit à obtenir des meilleurs gains en performance de traduction. Nos résultats confirment la qualité des 1.5M paires de phrases extraites, de sorte que nous pourrions abaisser la valeur du seuil de décision afin d'extraire des corpus d'une plus grande taille. Étant donné la nature hors domaine des paires d'articles de Wikipédia par rapport au domaine de l'ensemble d'entraînement utilisé pour entraîner notre système d'extraction de phrases parallèles (c'est-à-dire Europarl), ces résultats sont très encourageants. Nous voyons que notre approche pourrait bien s'appliquer avec des corpus comparables ayant un degré

Données	Modèle	BLEU		Paires
		SMT	**NMT**	
Europarl		21.47	17.63	500,000
		23.18	25.36	2,000,000
+Top250k	BiRNN	**23.11 (+1.64)**	**24.44 (+6.81)**	750,000
	Baseline	23.09 (+1.62)	24.22 (+6.59)	750,000
+Top500k	BiRNN	**24.12 (+2.65)**	**25.93 (+8.30)**	1,000,000
	Baseline	23.96 (+2.49)	25.82 (+8.19)	1,000,000
+Top750k	BiRNN	**24.53 (+3.06)**	**26.39 (+8.76)**	1,250,000
	Baseline	24.44 (+2.97)	26.19 (+8.56)	1,250,000
+Top1M	BiRNN	**24.85 (+3.38)**	**26.64 (+9.01)**	1,500,000
	Baseline	24.66 (+3.19)	26.59 (+8.96)	1,500,000
+Top1.25M	BiRNN	**25.01 (+3.54)**	**26.80 (+9.17)**	1,750,000
	Baseline	24.72 (+3.25)	26.64 (+9.01)	1,750,000
+Top1.5M	BiRNN	**25.18 (+3.71)**	**27.10 (+9.47)**	2,000,000
	Baseline	25.01 (+3.54)	26.85 (+9.22)	2,000,000

Tableau 4. IV. Scores BLEU obtenus sur l'ensemble de test newstest2013. Paires est le nombre de paires de phrases dans l'ensemble d'entraînement des systèmes de traduction automatique. Les lignes Europarl sont deux systèmes de références entraînés uniquement sur le corpus Europarl. Les nombres entre parenthèses sont les gains par rapport aux systèmes de référence entraînés avec 500k paires de phrases du corpus Europarl (première ligne du tableau).

de comparabilité plus faible. D'un autre côté, les bons résultats obtenus avec le système d'extraction de référence contredit d'une certaine manière l'observation de (Munteanu and Marcu, 2005) mentionnant que leur système réussit mal à généraliser sur des textes hors domaine. Cela pourrait s'expliquer par le fait que les paires d'articles Wikipédia dans deux langues pourraient être en général des textes comparables avec un degré de similarité relativement élevé.

Lorsque des corpus parallèles sont récoltés sur le Web, (Goutte et al., 2012) montre que les systèmes SMT sont tolérants au bruit retrouvé dans ces corpus. Les auteurs recommandent de ne pas nettoyer les corpus d'entraînement si l'erreur d'alignement est inférieur à 30%. Cependant, aucune étude n'a encore été faite pour supporter cette observation pour les systèmes NMT. Malgré que nous ne faisons pas une étude exhaustive sur cette question, nous voyons que les paires de phrases extraites des corpus comparables semblent être une avenue particulièrement prometteuse pour améliorer la performance des systèmes NMT. En effet, nous observons qu'ajouter les 250k paires de phrases les moins bien classées parmi les

1.5M paires de phrases extraites par BiRNN améliore le score BLEU de 26.80 à 27.10.

Des échantillons d'exemples de paires de phrases extraites de Wikipédia par les deux systèmes sont présentés dans les Tableau 4. V et 4. VI. Pour chaque système, nous avons sélectionné une paire de phrase par tranche de 250k observations de l'ensemble des 1.5M paires de phrases les mieux classées. Il est difficile d'en faire un jugement de manière qualitative, outre qu'observer que le niveau de parallélisme entre les paires de phrases sélectionnées est excellent. En fait, nous voyons que la dernière paire de phrases pour les deux systèmes est pratiquement parallèle. Cela dit, il serait possible d'ajuster le seuil de décision de manière à extraire davantage de paires de phrases de bonne qualité. Par ailleurs, nous voyons que les phrases couvrent des domaines et un vocabulaire différents que ceux des corpus parallèles standards, souvent tirés des procédures des sessions parlementaires et des textes de nouvelles journalistiques.

Top{0,250k}	AN	Statistical NLP comprises all quantitative approaches to automated language processing, including probabilistic modeling, information theory, and linear algebra.
	FR	Le TAL statistique comporte toutes les approches quantitatives du traitement linguistique automatisé, y compris la modélisation, la théorie de l'information, et l'algèbre linéaire.
Top{250k,500k}	AN	Feynman did not dispute the quark model ; for example, when the fifth quark was, discovered in 1977 Feynman immediately pointed out to his students that the discovery implied the existence of a sixth quark, which was duly discovered in the decade after his death.
	FR	Feynman ne remettait pas en cause le modèle des quarks, par exemple quand le 5 quark (bottom) fut découvert, Feynman annonça aussitôt à ses élèves que cette découverte impliquait l'existence d'un 6 quark (top) qui fut effectivement découvert dix ans après sa mort.
Top{500k,750k}	AN	Besides machine learning and neuroscience, other fields represented at NIPS include cognitive science, psychology, computer vision, statistical linguistics, and information theory.
	FR	D'autres domaines sont présents à NIPS, comme la science cognitive, la vision par ordinateur, la psychologie ou la théorie de l'information.
Top{750k,1M}	AN	He further predicted that machine learning would be an important part of building powerful machines, a claim considered plausible by contemporary researchers in artificial intelligence.
	FR	Il a aussi prédit que l'acquisition par apprentissage des ordinateurs serait aussi importante pour construire des ordinateurs performants, une affirmation qui est aujourd'hui considérée comme plausible par les chercheurs contemporains en intelligence artificielle.
Top{1M,1.25M}	AN	After attending a recital of Murray Perahia, where Perahia performed Chopin's Third Piano Sonata without observing the first movement repeat, Richter asked him backstage to explain the omission.
	FR	Ainsi, après avoir assisté à un récital de Murray Perahia, où Perahia avait interprété la troisième sonate pour piano de Chopin sans observer la répétition du premier mouvement, Richter lui demanda dans les coulisses de lui expliquer les raisons de cette omission.
Top{1.25M,1.5M}	AN	The arXiv (pronounced "archive", as if the "X" were the Greek letter "Chi", χ) is repository of electronic preprints of scientific papers in the fields of mathematics, physics, astronomy, computer science, quantitative biology, statistics, and quantitative finance, which can be accessed online.
	FR	arXiv (prononcé comme on prononce le « X » dans LaTeX ou TeX, c'est-à-dire comme ou) est une archive de "prépublications électroniques" d'articles scientifiques dans les domaines de la physique, l'astrophysique, des mathématiques, de l'informatique, des sciences non linéaires et de la biologie quantitative, et qui est accessible gratuitement par Internet.

Tableau 4. V. Exemples de paires de phrases en anglais (AN) et en français (FR) extraites de Wikipédia par BiRNN.

Top{0,250k}	AN	New Caledonia lies at the northern end of the ancient continent, while New Zealand rises at the plate boundary that bisects it.
	FR	La Nouvelle-Calédonie se situe à l'extrémité nord de l'ancien continent, tandis que la Nouvelle-Zélande s'élève à la limite des plaques qui la coupe en deux.
Top{250k,500k}	AN	Horzine Biotech, a biotechnology company, is contracted to conduct experiments of a military nature involving mass cloning and genetic manipulation.
	FR	À Londres, en Angleterre, Horzine Biotech, une compagnie de biotechnologie, mène des expériences militaires secrètes sur le clonage de masse et les manipulations génétiques.
Top{500k,750k}	AN	Hansen's best known contribution to economics was his and John Hicks' development of the IS-LM model, also known as the Hicks-Hansen synthesis.
	FR	La contribution d'Hansen la plus connue à la théorie économique est l'apport à l'élaboration du modèle IS/LM aussi connu sous le nom de synthèse Hicks-Hansen.
Top{750k,1M}	AN	In the medieval Arab Agricultural Revolution, a social transformation took place as a result of changing land ownership giving individuals of any gender, ethnic or religious background the right to buy, sell, mortgage and inherit land.
	FR	À l'époque de la révolution agricole arabe, une transformation sociale eut lieu en conséquence du changement de la politique de propriété des terres : toute personne, quels que soient son sexe, son ethnie ou sa religion, eut le droit de vendre, d'acheter, d'hériter et d'hypothéquer une terre.
Top{1M,1.25M}	AN	The interplay of fire, water, destruction and other natural forces provides a constant accompaniment to the events of the novel, as do light and darkness, noise and silence, sun and moon, storms and tranquility, and other powerful polarities.
	FR	L'interaction des éléments et des forces naturels, tels que le feu, l'eau et la destruction, est un élément essentiel du roman, auquel il convient d'ajouter les puissants contrastes entre la clarté et les ténèbres, le bruit et le silence, le soleil et la lune, les orages et la tranquillité.
Top{1.25M,1.5M}	AN	Paul obtained his bachelor s degree at the University of Wisconsin-Oshkosh in 1990.
	FR	Brian a obtenu le diplôme de l'université du Wisconsin-Oshkosh en 1990.

Tableau 4. VI. Exemples de paires de phrases en anglais (AN) et en français (FR) extraites de Wikipédia par le système de référence.

Chapitre 5

CONCLUSION

Ce mémoire explore le manque de ressources parallèles qui constitue un problème pour le développement des applications multilingues du TALN. Pour résoudre ce problème, nous proposons un nouveau système d'extraction de phrases parallèles servant à construire des corpus parallèles à partir d'une collection de textes dans plusieurs langues. Les systèmes développés jusqu'à présent étaient basés sur l'utilisation d'une série de modèles qui demandait de mesurer plusieurs *features* qui ne s'adaptaient pas nécessairement bien aux différentes structures des textes. Pour alléger ce problème, nous proposons de tirer profit des avancées en apprentissage profond en utilisant un système à base de réseaux de neurones récurrents bidirectionnels qui traite directement des paires de phrases afin de déterminer si deux phrases sont une traduction l'une de l'autre. Notre approche peut être considérée comme un système caractérisant une paire de phrase en représentations vectorielles continues pour mesurer leur similarité interlangue.

Les résultats de nos expériences illustrent l'utilité de notre approche. Dans une première expérience, où nous extrayons des paires de phrases à partir de corpus parallèles dans lesquels des phrases non parallèles ont été ajoutées, nous montrons que notre approche obtient des résultats significatement supérieurs à ceux d'un système de référence considéré comme l'état de l'art. Nos systèmes ont ensuite été mis en valeur en extrayant des paires de phrases à partir d'articles anglais-français de Wikipédia pour mesurer l'impact qu'elles ont sur la perfomance de traduction des systèmes de traduction automatique SMT et NMT. Les résultats obtenus démontrent que les systèmes de traduction exploitant les phrases extraites de notre approche performent mieux que ceux utilisant celles du système de référence. Pour la première fois, nous avons trouvé qu'un corpus parallèle contenant jusqu'à 1.5M nouvelles paires de phrases extraites d'un corpus comparable est une ressource considérable à exploiter pour les systèmes NMT. Ces observations montrent que notre approche à la capacité d'extraire plusieurs autres millions de paires de phrases en utilisant les ressources

retrouvées sur le Web pour enrichir le développement des applications multilingues du TALN.

C'est la première fois qu'un système d'extraction de phrases parallèles est complétement basé sur des modèles à base de réseaux de neurones et nous croyons que notre approche a beaucoup de potentiel comme pistes de travaux futurs. Nous les mentionnons brièvement ci-dessous.

Dans le cadre de notre travail, nous n'avons pas traité les mots inconnus et il est connu que cela est problématique pour le bon fonctionnement de notre approche sur des textes ayant un vocabulaire différent à celui des données d'entraînement. Pour répondre à ce problème, lors du développement des vocabulaires il pourrait être avantageux d'utiliser les caractères (Chung et al., 2016) ou des segments de mots (Sennrich et al., 2015) comme unités lexicales au lieu des mots.

Le principal enjeu avec notre approche (et la très grande majorité des systèmes d'extraction de phrases parallèles) est que nous avons besoin d'un corpus parallèle pour faire l'apprentissage des paramètres de notre modèle. Cela est comme le paradoxe de l'oeuf et de la poule : c'est-à-dire que nous avons besoin d'un ensemble de paires de phrases parallèles pour faire l'extraction de paires de phrases parallèles. Bien qu'il serait possible de partir d'un petit corpus parallèle et d'itérer l'apprentissage des paramètres par *bootstrap* avec les nouvelles paires de phrases extraites des corpus comparables, cette méthode est difficilement applicable pour les nombreuses paires de langues ayant peu de ressources disponibles. Pour réduire le besoin de la quantité des données parallèles, une avenue très intéressante et prometteuse serait d'adapter notre système avec des méthodes utilisées dans les approches (Irvine and Callison-Burch, 2016; Xia et al., 2016; Lample et al., 2017) décrites à la section 2.3. Par ailleurs, en employant du *adversarial learning*, notre modèle pourrait peut-être servir comme discriminateur pré-entraîné servant à entraîner conjointement un générateur de phrases parallèles qui cherche à le tromper. En effet, il s'offre une vaste étendue de possibilités pour élargir la capacité du système avec des architectures plus sophistiquées ou d'autres mesures de similarité.

Au lieu de sélectionner uniquement des exemples négatifs aléatoirement, il devrait être avantageux de s'inspirer du processus de filtrage des systèmes traditionnels pour sélectionner des exemples négatifs moins naïfs. De cette manière, les exemples négatifs d'un ensemble d'entraînement pourraient être constitués d'un mélange de paires de phrases aléatoires et plus difficiles (c'est-à-dire des paires de phrases non parallèles similaires).

Même si nous soutenons que notre approche est facilement extensible sur plusieurs paires de langues, ce mémoire a seulement évalué notre approche sur des textes anglais-français. Par conséquent, il serait intéressant d'étudier un ensemble de paires de langues plus large. De plus, nous soulignons qu'il serait très pertinent d'évaluer notre approche avec des paires de langues distantes ayant peu de ressources disponibles.

Finalement, durant nos expériences, notre approche classait indépendamment chaque paire de phrases du produit cartésien comme étant parallèle lorsque la probabilité obtenue était supérieure ou égale au seuil de décision. Cela pouvait souvent mener à une situation où une phrase source était associée à plusieurs phrases cibles, ou vice versa. Comme étape de post-traitement, nous pouvons améliorer la précision du système en garantissant que les phrases dans les deux langues apparaissent au plus dans une seule paire de phrases (c'est-à-dire obtenir un alignement un à un). Une technique simple et efficace pour y arriver est de trier par ordre décroissant les paires de phrases extraites en fonction du score de probabilité et d'itérer de manière vorace sur cette séquence en éliminant les paires dont la phrase source ou cible a déjà été associée à une paire de phrases.

Références

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. USENIX Association, Berkeley, CA, USA, OSDI'16, pages 265–283. http://dl.acm.org/citation.cfm?id=3026877.3026899.

Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the use of comparable corpora to improve smt performance. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, EACL '09, pages 16–23. http://dl.acm.org/citation.cfm?id=1609067.1609068.

Sisay Adafre and Maarten de Rijke. 2006. Finding similar sentences across multiple languages in wikipedia. pages 62–69.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Alberto Barrón-Cedeño, Cristina España-Bonet, Josu Boldoba, and Lluís Màrquez. 2015. A factory of comparable corpora from wikipedia. Association for Computational Linguistics, pages 3–13. https://doi.org/10.18653/v1/W15-3402.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.* 3 :1137–1155. http://dl.acm.org/citation.cfm?id=944919.944966.

Yoshua. Bengio, Patrice. Simard, and Paolo. Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Transactions on Neural Networks* 5(2) :157–166. https://doi.org/10.1109/72.279181.

Alexandre Bérard. 2014. Better handling of a bilingual collection of texts. MSc thesis, Université de Montréal.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR* abs/1607.04606. http://arxiv.org/abs/1607.04606.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, NIPS'93, pages 737–744. http://dl.acm.org/citation.cfm?id=2987189.2987282.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.* 19(2) :263–311. http://dl.acm.org/citation.cfm?id=972470.972474.

Jiang Chen and Jian-Yun Nie. 2000. Parallel web text mining for cross-language ir. In *Content-Based Multimedia Information Access - Volume 1*. Le Centre De Hautes Études Internationales d'Informatique Documentaire, Paris, France, France, RIAO '00, pages 62–77. http://dl.acm.org/citation.cfm?id=2835865.2835872.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Comput. Speech Lang.* 13(4) :359–394. https://doi.org/10.1006/csla.1999.0128.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078. http://arxiv.org/abs/1406.1078.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2016. Parallel sentence extraction from comparable corpora with neural network features. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Paris, France.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. *CoRR* abs/1603.06147. http://arxiv.org/abs/1603.06147.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '08, pages 160–167. https://doi.org/10.1145/1390156.1390177.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *CoRR* abs/1710.04087. http://arxiv.org/abs/1710.04087.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science* 14(2) :179–211. https://doi.org/10.1016/0364-0213(90)90002-E.

William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Comput. Linguist.* 19(1) :75–102. http://dl.acm.org/citation.cfm?id=972450.972455.

Cyril Goutte, Marine Carpuat, and Georges Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *The Association for Machine Translation in the Americas 2012*. AMTA '12. http://www-labs.iro.umontreal.ca/ foster/papers/align-error-amta12.pdf.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. JMLR.org, ICML'15, pages 748–756. http://dl.acm.org/citation.cfm?id=3045118.3045199.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR* abs/1308.0850. http://arxiv.org/abs/1308.0850.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2015. LSTM: A search space odyssey. *CoRR* abs/1503.04069. http://arxiv.org/abs/1503.04069.

Francis Grégoire and Philippe Langlais. 2017. A deep neural network approach to parallel sentence extraction. *CoRR* abs/1709.09783. http://arxiv.org/abs/1709.09783.

Kenneth Heafield. 2011. Kenlm : Faster and smaller language model queries. In *In Proc. of the Sixth Workshop on Statistical Machine Translation*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pages 1693–1701. http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Compututation* 9(8) :1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Ann Irvine and Chris Callison-Burch. 2016. End-to-end statistical machine translation with zero or small parallel texts. *Natural Language Engineering* 22(4) :517–548. https://doi.org/10.1017/S1351324916000127.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. JMLR.org, ICML'15, pages 2342–2350. http://dl.acm.org/citation.cfm?id=3045118.3045367.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *CoRR* abs/1701.02810. http://arxiv.org/abs/1701.02810.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '07, pages 177–180. http://dl.acm.org/citation.cfm?id=1557769.1557821.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL '03, pages 48–54. https://doi.org/10.3115/1073445.1073462.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR* abs/1711.00043. http://arxiv.org/abs/1711.00043.

Fethi Lamraoui and Philippe Langlais. 2013. Yet another fast, robust and open source sentence aligner. time to reconsider sentence alignment ? In *XIV Machine Translation Summit*. Nice, France.

Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. 1998. Effiicient backprop. In *Neural Networks : Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*. Springer-Verlag, London, UK, UK, pages 9–50. http://dl.acm.org/citation.cfm?id=645754.668382.

Adam Lopez. 2008. Statistical machine translation. *ACM Comput. Surv.* 40(3) :8 :1–8 :49. https://doi.org/10.1145/1380584.1380586.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *NAACL Workshop on Vector Space Modeling for NLP*. Denver, United States.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781. http://arxiv.org/abs/1301.3781.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model 2 :1045–1048.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR* abs/1309.4168. http://arxiv.org/abs/1309.4168.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation : From Research to Real Users*. Springer-Verlag, London, UK, UK, AMTA '02, pages 135–144. http://dl.acm.org/citation.cfm?id=648181.749407.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics* 31(4) :477–504. https://doi.org/10.1162/089120105775299168.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.* 29(1) :19–51. https://doi.org/10.1162/089120103321337421.

Pablo Gamallo Otero and Isaac González López. 2010. Wikipedia as multilingual source of comparable corpora.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02, pages 311–318. https://doi.org/10.3115/1073083.1073135.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. JMLR.org, ICML'13, pages III–1310–III–1318. http://arxiv.org/abs/1211.5063.

Alexandre Patry and Philippe Langlais. 2011. Identifying parallel documents from a large bilingual collection of texts: Application to parallel article extraction in wikipedia. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora : Comparable Corpora and the Web*. Association for Computational Linguistics, Stroudsburg, PA, USA, BUCC '11, pages 87–95. http://dl.acm.org/citation.cfm?id=2024236.2024252.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *CoRR* abs/1609.04747. http://arxiv.org/abs/1609.04747.

Sebastian Ruder. 2017. A survey of cross-lingual embedding models. *CoRR* abs/1706.04902. http://arxiv.org/abs/1706.04902.

David. E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. MIT Press, Cambridge, MA, USA, chapter Learning Internal Representations by Error Propagation, pages 318–362. http://dl.acm.org/citation.cfm?id=104279.104293.

Mike. Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.* 45(11) :2673–2681. https://doi.org/10.1109/78.650093.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR* abs/1508.07909. http://arxiv.org/abs/1508.07909.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '10, pages 403–411. http://dl.acm.org/citation.cfm?id=1857999.1858062.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *CoRR* abs/1702.03859. http://arxiv.org/abs/1702.03859.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*. Association for Computational Linguistics, pages 196–205. https://doi.org/10.3115/v1/N15-1020.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, NIPS'14, pages 3104–3112. http://dl.acm.org/citation.cfm?id=2969033.2969173.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '10, pages 384–394. http://dl.acm.org/citation.cfm?id=1858681.1858721.

Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pages 1101–1109. http://dl.acm.org/citation.cfm?id=1873781.1873905.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '08, pages 1096–1103. https://doi.org/10.1145/1390156.1390294.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR* abs/1506.05869. http://arxiv.org/abs/1506.05869.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR* abs/1411.4555. http://arxiv.org/abs/1411.4555.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '96, pages 836–841. https://doi.org/10.3115/993268.993313.

Yingce Xia, Di He, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. *CoRR* abs/1611.00179. http://arxiv.org/abs/1611.00179.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *CoRR* abs/1502.03044. http://arxiv.org/abs/1502.03044.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *CoRR* abs/1412.1632. http://arxiv.org/abs/1412.1632.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR* abs/1409.2329. http://arxiv.org/abs/1409.2329.